Міністерство освіти і науки України
Міжнародний економіко-гуманітарний університет
імені академіка Степана Дем'янчука
Кафедра математичного моделювання

Н.С.ПАНЧЕЛЮГА

ПОБУДОВА І ДОСЛІДЖЕННЯ ЕКОНОМІКО-МАТЕМАТИЧНОЇ МОДЕЛІ ЗАЛЕЖНОСТІ ЦІНИ ТОВАРУ І ПОПИТУ МЕТОДОМ СТАТИСТИЧНИХ ВИПРОБУВАНЬ МОНТЕ-КАРЛО

МОДЕЛЬ 81КІН-М42

Науковий керівник:
Р.М.Літнарович, доцент,
кандидат технічних наук

Рівне 2010

УДК 519.87

Панчелюга Н.С. Побудова і дослідження економіко-математичної моделі залежності ціни товару і попиту методом статистичних випробувань Монте-Карло, модель 81КІН-М42, науковий керівник Р.М.Літнарович.МЕГУ, Рівне, 2010, 111с.

На основі фактичних даних залежності ціни товару і попиту на нього побудована математична модель у вигляді кубічного поліному за способом найменших квадратів.

В даній роботі генеруються середні квадратичні похибки, які приводяться до заданих нормованих, будується спотворена модель, зрівноважується за способом найменших квадратів. Знаходяться ймовірніші значення коефіцієнтів *а, в, с, d* кубічного поліному апроксимуючої математичної моделі.

Робиться оцінка точності і даються узагальнюючі висновки.

Застосований метод статистичних випробувань Монте Карло дав можливість провести широкомасштабні дослідження і набрати велику статистику.

Для студентів і аспірантів Економічного факультету і факультету Кібернетики МЕГУ.

Panchelyuga N.S. Construction and research of ekonomiko mathematical model of dependence of cost of commodity and demand by the method of statistical tests of Monte Carlo, model 81KIN-M42, IEGU, Rivne, 2010, 111p.

On the basis of fact sheets of dependence of cost of commodity and demand on him a mathematical model is built as cube a polynomial on the method of leastsquares.

Middle quadratic errors which over are brought to set rationed are generated in this work, the disfigured model is built, counterbalanced on the method of leastsquares. There are more credible values of coefficients *a,b,c, d* cube the polynomial of approximating mathematical model.

The estimation of exactness is done and summarizings are given conclusions.

The method of statistical tests of Monte Karlo is applied enabled to conduct large-scale researches and collect large statistics.

For students and graduate students of the Economic faculty and faculty of Cybernetics IEGU.

Рецензент: В. Г. Бурачек, доктор технічних наук, професор

Відповідальний за випуск: Й. В. Джунь, доктор фізико-математичних наук, професор

Книга написана за матеріалами роботи наукової фізико-математичної школи МЕГУ

ЗМІСТ

ВСТУП

Відтоді як економіка стала серйозною самостійною наукою, дослідники намагаються спрогнозувати ту чи іншу ситуацію, передбачити майбутні значення економічних показників, запропонувати інструменти зміни ситуації в бажаному напрямку. Одним з головних напрямів розвитку економіки є застосування ефективних наукових методів аналізу й оптимізації складних економіко-організаційних систем. Серед наукових методів, які застосовуються в економіці, науці і техніці, особливе місце займають методи моделювання. Моделювання- це процес створення (і дослідження) моделі. Під моделлю деякого об’єкта розуміється сукупність найважливіших властивостей об’єкта процесу чи явища, яка володіє суттєвими для цілей моделювання властивостями і в рамках цих цілей повністю замінює вихідний об’єкт, процес чи явище.

Найзагальнішим методом дослідження, та таким що найбільше використовується в науці, зокрема в кібернетиці, можна назвати процес створення математичної моделі, тобто математичне моделювання. Неможливо уявити собі сучасну науку без широко застосування математичного моделювання, суть якого полягає в заміні досліджуваного об’єкта його "образом" - математичною моделлю – і подальшому вивченні моделі за допомогою відповідних обчислювально-логічних алгоритмів на ЕОМ. Робота не з об’єктом (явищем, процесом), а з його моделлю дає можливість без істотних затрат і відносно швидко дослідити його властивості і поведінку у різних ситуаціях. Обчислювальні (комп’ютерні, стимуляційні, імітаційні) експерименти з моделями об’єктів дозволяють детально вивчати об’єкти з достатньою повнотою, недоступної для чисто теоретичних досліджень. В економіці математичне моделювання застосовують при вивченні складних економічних явищ. Моделі що будуються при цьому називають економіко-математичними. Економіко- математична модель (ЕММ) - це опис, що відображає економічний процес або явище за допомогою одного або декількох математичних виразів (рівнянь, функцій, нерівностей, тотожності), що імітують поведінку модельованого об'єкту в заданих або можливих умовах

його реального існування. Економіко-математичне моделювання, будучи одним з системних методів дослідження, дозволяє у формалізованій формі визначити причини, чому змінюються економічні явища, закономірності цих змін, їх наслідки, а також робить можливим прогнозування економічних процесів.

При побудові моделей ті або інші вірогідні ситуації або гіпотези фахівців стають більш осяжними, можуть уточнюватися, а тому сприяють кращому розумінню ситуації. Моделювання прискорює підготовку рішень і страхує від грубих помилок в діяльності підприємства. Застосування ЕОМ при моделюванні дозволяє імітувати багатоваріантні ситуації, які можуть скластися на ринках збуту, матеріально-технічного забезпечення або усередині структур підприємства, що для економічних процесів, де небажане будь- яке експериментування є досить важливим.

Одним з імітаційних методів є метод Монте- Карло. Цей метод дозволяє моделювати будь-який процес, на протікання якого впливають випадкові чинники. Ідея цього методу: якщо нам треба приблизно вирахувати деяку величину А, то треба придумати таку випадкову величину В, що отримавши і обробивши множину її значень можна було отримати шукану величину. Для багатьох математичних завдань, не пов'язаних з якими-небудь випадковостями, можна штучно придумати імовірнісну модель, яка в деяких випадках є вигіднішою. Оскільки метод Монте-Карло вимагає проведення великого числа випробувань, його часто називають методом статистичних випробувань. Метод Монте- Карло – могутній і універсальний інструмент для розв”язку задач в багатьох областях знань.

Метою даної роботи є побудова та дослідження за результатами фактичних даних економіко- математичної моделі залежності попиту на товар від його ціни методом статистичних випробувань Монте- Карло та розробка і створення програми у середовищі Delphi. В першому розділі описуються особливості економіко- математичного моделювання, класифікація економічних моделей та етапи створення цих моделей, теоретичні основи імітаційного моделювання та метод статистичних випробувань, як один з методів імітаційного моделювання . В другому розділі описуються випадкові числа, способи їх отримання та генератори випадкових чисел.

В третьому та четвертому розділі генеруються середні квадратичні похибки, які приводяться до заданих нормованих, будується спотворена модель, зрівноважується за способом найменших квадратів. Знаходяться ймовірніші значення коефіцієнтів *a, в, c, d* кубічного поліному апроксимуючої математичної моделі. В п”ятому розділі робиться оцінка точності елементів зрівноважених процедурою способу найменших квадратів. В шостому розділі йде опис самої програми та її компонентів.

РОЗДІЛ 1. ЕКОНОМІКО – МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ.

1.1 Поняття моделі та моделювання.

Модель – речова, знакова або уявна (мислена) система, що відтворює, імітує, відображає принципи внутрішньої організації або функціонування, певні властивості, ознаки чи характеристики об’єкта дослідження (оригіналу).

Значення терміна “модель” багатопланове:

– зразок, взірцевий примірник чогось;
– тип, марка конструкції;
– те, що є матеріалом, натурою для відтворення;
– зразок, з якого знімається форма для відливання в іншому матеріалі;
– комп’ютерна модель,
– розрахункова модель,
– теоретична модель (процесу, конструкції тощо).

Розрізняють фізичні, математичні та ін. моделі.

Наприклад, модель — опис об'єкта (предмета, явища або процесу) на якій-небудь формалізованій мові, складений з метою вивчення його властивостей. Такий опис особливо корисний у випадках, коли дослідження самого об'єкта ускладнене або фізично неможливе.

Найчастіше в ролі моделі виступає інший матеріальний або уявний об'єкт, що замінює в процесі дослідження об'єкт-оригінал. Таким чином, модель виступає як своєрідний інструмент для пізнання, який дослідник ставить між собою і об'єктом, і за допомогою якого вивчає об'єкт, що його цікавить.

Математична модель — це система математичних співвідношень, які описують досліджуваний процес або явище. Математична модель має важливе значення для таких наук, як:

економіка, екологія, соціологія, фізика, хімія, механіка, інформатика, біологія та ін.

При одержанні математичної моделі використовують загальні закони природознавства, спеціальні закони конкретних наук, результати пасивних та активних експериментів, імітаційне моделювання за допомогою ЕОМ.

Математичні моделі дозволяють передбачити хід процесу, розрахувати цільову функцію (вихідні параметри процесу), керувати процесом, проектувати системи з бажаними характеристиками. Для їх створення можна використовувати будь які математичні засоби — мову диференційних або інтегральних рівнянь, теорії множин, абстрактної алгебри, математичну логіку, теорії ймовірностей, графи та інші.

Якщо відношення задаються аналітично, то їх можна розв'язати в замкнутому вигляді (явно) відносно шуканих змінних як функції від параметрів моделі, або в частково замкнутому вигляді (неявно), коли шукані змінні залежать від одного або багатьох параметрів моделі. До моделей цього класу належать диференційні, інтегральні, різницеві рівняння, ймовірнісні моделі, моделі математичного програмування та інші. Якщо не можна здобути точний розв'язок математичної моделі, використовуються чисельні (обчислювальні) методи або інші види моделювання.

У залежності від того, якими є параметри системи та зовнішні збурення математичні моделі можуть бути детермінованими та стохастичними. Детерміновані- пов"язані з дослідженням моделей з чітко заданими параметрами (задання початкових і граничних умов значень функцій на вході і т. д.). Стохастичні – пов"язані з випадковими значеннями. Останні мають особливо важливе значення при дослідженні і проектуванні великих систем зі складними зв'язками і властивостями, які важко врахувати.

Для розробки математичних моделей широко використовується диференційне числення, теорія множин, матриці і графи, а також планування експерименту. Відповідно розрізняють теоретико-множинні, матричні, топологічні та поліномні математичні моделі.

Приклади математичних моделей:

1. Модель Мальтуса – закон про пропорційну залежність

між швидкістю росту і розміром популяції.

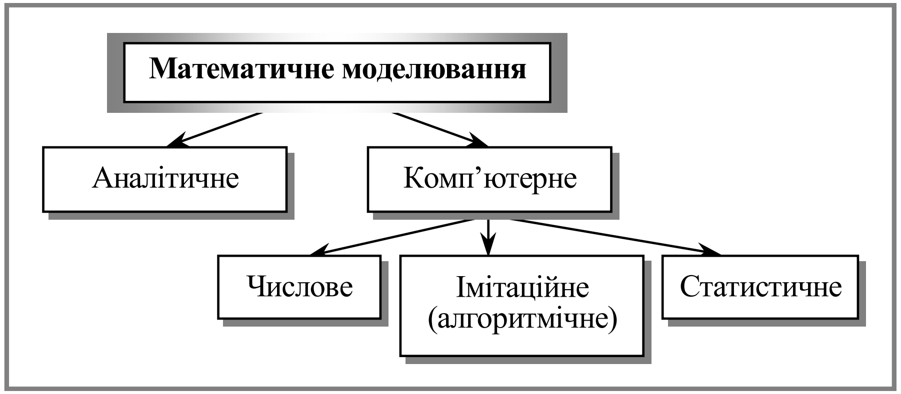
2. Система хижак-жертва (Вольтера-Лотки) – показує залежність між чисельністю хижаків та жертв.

3. Модель оптимальної поведінки покупця – виражає вибір покупця між множиною продуктів при обмеженому бюджеті.

Процес побудови, вивчення й використання математичних моделей називається математичним моделюванням. Це найзагальніший та найбільш використовуваний в науці, зокрема, в кібернетиці, метод досліджень. Це метод дослідження процесів або явищ шляхом створення їхніх математичних моделей і дослідження цих моделей. Він тісно поєднаний з такими категоріями, як абстракція, аналогія, гіпотеза тощо.В основу методу покладено ідентичність форми рівнянь і однозначність співвідношень між змінними в рівняннях оригіналу і моделі, тобто, їх аналогії. Математичні моделі досліджуються, як правило, із допомогою аналогових обчислювальних машин, цифрових обчислювальних машин, комп'ютерів.

Математичне моделювання тією чи іншою мірою застосовують всі природничі і суспільні науки, що використовують математичний апарат для одержання спрощеного опису реальності за допомогою математичних понять. Воно дозволяє замінити реальний об'єкт його моделлю і потім вивчати останню.

В залежності від характеру процесів, що вивчаються, в системі всі види моделювання можуть бути розділені на аналітичні та комп"ютерні (мал.1)



Мал.1. Аналітичне та комп'ютерне моделювання

Для аналітичного моделювання характерним є те, що процеси функціонування елементів системи записують у вигляді деяких математичних співвідношень (алгебраїчних, інтегро-диференційних, кінцево-різницевих тощо) чи логічних умов.

Аналітична модель може досліджуватися такими методами:

1. Аналітичним, коли прагнуть у загальному вигляді отримати деякі залежності для шуканих характеристик;
2. Числовим;
3. Якісним, коли, не маючи явного розв'язку, все ж знаходять деякі властивості рішень.

Комп'ютерне моделювання характеризується тим, що математична модель системи (використовуючи основні співвідношення аналітичного моделювання) подається у вигляді деякого алгоритму та програми, придатної для її реалізації на комп'ютері, що дозволяє проводити з нею обчислювальні експерименти. Залежно від математичного інструментарію, що використовується в побудові моделі, та способу організації обчислювальних експериментів можна виокремити три взаємопов'язані види моделювання: числове, алгоритмічне (імітаційне) та статистичне.

За числового моделювання для побудови комп'ютерної моделі використовуються методи обчислювальної математики.

Алгоритмічне (імітаційне) моделювання (може бути як детермінованим, так і стохастичним) — це вид комп'ютерного моделювання, для якого характерним є відтворення на комп'ютері (імітація) процесу функціонування досліджуваної складної системи. Тут імітуються (з використанням аналітичних залежностей і моделей) елементарні явища, що становлять процес, зі збереженням їхньої логічної та семантичної структури, послідовності плину в часі, що дозволяє отримати нову інформацію про стан системи S у задані моменти часу.

Статистичне моделювання — це вид комп'ютерного моделювання, який дозволяє отримати статистичні дані відносно процесів у модельованій системі S.

Все частіше використовується комбіноване моделювання Системотвірним елементом якого є аналітичні моделі.У побудові

та використанні комбінованих моделей попередньо проводять декомпозицію процесу функціонування моделі на складові елементи.

З розвитком математичних досліджень ускладнюється й проблема класифікації моделей, що використовуються. Разом із виникненням нових типів моделей (особливо змішаних типів) і нових ознак їх класифікації здійснюється процес інтеграції моделей різних типів у більш складні модельні конструкції.

Значне місце серед математичних моделей займають економіко-математичні моделі.

1.2. Математична модель економіки.

При вивченні складних економічних процесів та явищ часто застосовується моделювання. Модель- це спеціально створений об'єкт, на якому відтворюються певні характеристики досліджуваного явища, а моделювання- це конкретне відтворення цих характеристик, що дає змогу вивчати можливу поведінку явища без проведення експериментів над ним. Для економіки, де неможливе будь-яке експериментування, особливого значення набуває математичне моделювання. Завдяки застосуванню потужного математичного апарату воно є найефективнішим і найдосконалішим методом.

Математичну модель, що описує механізм функціонування певної економічної чи соціально-економічної системи називають економіко-математичною чи просто економічною. Економіко-математична модель- це математичний опис економічного явища чи об'єкта, який виконується з метою дослідження цього явища чи об"єкта, та управління ним.

Математична модель, аби бути ефективним інструментом вивчення економічних процесів, насамперед має відповідати таким вимогам: будуватися на основі економічної теорії й відбивати об'єктивні закономірності процесів; правильно відтворювати функцію та структуру реальної економічної системи; відповідати певним математичним умовам (мати розв'язок, узгоджені розмірності тощо). Результати досліджень будь-якої моделі можуть мати практичну цінність, якщо модель адекватна явищу, що вивчається, тобто досить добре відтворює реальну ситуацію.

Побудова економіко-математичної моделі полягає в тому, щоб узгоджувати якомога більшу лаконічність у її математичному описі з достатньою адекватністю та точністю модельного відтворення тих сторін аналізованої економічної реальності, які, власне, і цікавлять дослідника згідно з цілями та взятими гіпотезами. [1,стр.56]

Моделювання економіки як науковий напрям сформувався у 60-ті роки XX століття, хоча має давню передісторію. У його основу, крім економічних, покладено низку фундаментальних дисциплін (математику, теорію ймовірностей, теорію систем, інформатику, статистику, теорію автоматичного управління тощо). Під економіко-математичною моделлю розуміють концентроване вираження найсуттєвіших економічних взаємозв'язків досліджуваних об'єктів (процесів) у вигляді математичних функцій, нерівностей і рівнянь.

Для класифікації цих моделей використовують різні класифікаційні ознаки.

За цільовим призначенням економіко-математичні моделі поділяються на теоретико-аналітичні та прикладні. Теоретико-аналітичні використовуються під час дослідження загальних властивостей і закономірностей економічних процесів, прикладні застосовуються у розв'язанні конкретних економічних задач (моделі економічного аналізу, прогнозування, управління).

Можна виокремити такі етапи економіко- математичного моделювання:

1. Постановка економічної проблеми та її якісний аналіз.

На цьому етапі формулюється сутність проблеми. Він включає виокремлення найважливіших рис і властивостей об'єкта, що моделюється; вивчення структури об'єкта і головних залежностей, що поєднують його елементи; формулювання гіпотез, що пояснюють поведінку і розвиток об'єкта.

2. Побудова математичних моделей.

На цьому етапі економічна проблема формалізується, здійснюється вираження її у вигляді конкретних математичних залежностей і відношень (функцій, рівнянь, нерівностей тощо).

3. Математичний аналіз моделі**.**

Метою цього етапу є з'ясування загальних властивостей моделі. Найважливіший момент — доведення існування рішень у сформованій моделі (теорема існування).

4. Підготовка вихідної інформації.

У процесі підготовки інформації широко використовуються методи теорії ймовірностей, теоретичної і математичної статистики. У статистичному економіко-математичному моделюванні результуюча інформація, яка використовується в одних моделях, є вихідною для функціонування інших моделей.

5. Числові розв'язки.

Цей етап включає розробку алгоритмів для числового розв'язування задачі, складання програм на ЕОМ і безпосереднє проведення розрахунків. Дослідження, які проводяться за допомогою числових методів, можуть стати суттєвим доповненням до результатів аналітичного дослідження. Клас економічних задач, які можна розв'язувати числовими методами, значно ширший, ніж клас задач, доступних аналітичному дослідженню.

6. Аналіз числових результатів та їх використання.

На цьому етапі проводять математичні методи перевірки правильності і повноти результатів моделювання, рівень практичного застосування цих результатів.

ЕММ можуть призначатися для дослідження різних сторін функціонування народного господарства.

Прикладами економічних моделей є моделі споживчого вибору, моделі фірми, моделі економічного зростання, моделі рівноваги на товарних, факторних і фінансових ринках тощо. Приведемо варіант так званої «павутиноподібної моделі», яка описує процес формування попиту і пропозиції певного товару чи виду послуг на ринку. Тобто, ідеться про формалізацію економічного закону попиту та пропозиції, згідно з яким:

1. Кількість товару, що його можна продати на ринку (тобто попит), змінюється у напрямку, протилежному зміні ціни товару;

2. Кількість товару, який виробляють і доставляють на ринок (тобто пропозиція), змінюється у тому самому напрямку, що й ціна;

3. Реальна ринкова ціна формується на рівні, на якому попит і пропозиція наближено дорівнюють одне одному, тобто перебувають у рівновазі.

Французький учений А. Курно, був першим, хто спробував математично сформулювати цей закон.

Математичні методи та моделі дозволяють упорядковувати систему економічної інформації, виявляти недоліки в наявній інформації і виробляти вимоги до підготовки нової інформації чи її коригування. Формалізація економічних задач і застосування комп'ютерів багаторазово прискорюють типові, масові розрахунки, підвищують точність і скорочують трудомісткість, дозволяють проводити багатоваріантні економічні дослідження. Завдяки застосуванню ЕММ значно підсилюються можливості конкретного кількісного аналізу, вивчення багатьох чинників, які впливають на економічні процеси. За допомогою математичного моделювання вдається розв'язувати такі економічні задачі, які іншими засобами розв'язати практично неможливо. Особливо це стосується задач, де суттєве значення мають стохастичні (випадкові) фактори, які виявляються у впливі на економіку як з боку природи та суспільства, так і у внутрішньоекономічних зв'язках. Такі задачі добре розв"язуються імітаційними (алгоритмічними) методами.

1.3. Імітаційне моделювання та метод статистичних випробувань.

Існує клас об'єктів, для яких з різних причин не розроблені аналітичні моделі або не розроблені методи розв'язування задач про такі моделі. В цьому випадку математична модель замінюється імітатором або імітаційною моделлю.

Імітаційна модель — логіко-математичний опис об'єкту, який може бути використаний для експериментування на комп'ютері в цілях проектування, аналізу і оцінки функціонування об'єкту. Імітаційне моделювання — це окремий випадок математичного моделювання, метод дослідження, заснований на тому, що система, яка вивчається, замінюється імітатором і з ним проводяться експерименти з метою отримання інформації про цю систему. Експериментування з імітатором називають імітацією (імітація — це збагнення суті явища, не вдаючись до експериментів на реальному об'єкті).

Імітаційне моделювання (машинна імітація) - особлива форма проведення експериментів на ЕОМ з математичними моделями, які з певним ступенем ймовірності описують закономірності функціонування реальних систем і об'єктів. Це метод, що дозволяє будувати моделі процесів, що описують, як ці процеси проходили б насправді. Таку модель можна «програти» в часі як для одного випробування, так і заданої їх кількості. При цьому результати визначатимуться випадковим характером процесів. За цими даними можна отримати достатньо стійку статистику.

Імітація як метод розв'язування нетривіальних задач отримала початковий розвиток у зв'язку із створенням ЕОМ в 1950х — 1960х роках.

Імітаційні (алгоритмічні) моделі можуть бути детермінованими і стохастичними. В останньому випадку за допомогою датчиків (генераторів) випадкових чисел імітується вплив (дія) невизначених і випадкових чинників. Такий метод імітаційного моделювання дістав назву методу статистичного моделювання (статистичних прогонів, чи методу Монте-Карло). На даний час цей метод вважають одним із найефективніших методів дослідження складних систем, а часто і єдиним практично доступним методом отримання нової інформації щодо поведінки гіпотетичної системи (на етапі її проектування).

Винахідником методу Монте-Карло називають Станіслава Улама (Stanislaw Ulam), американського математика, який народився у м. Львові. С.Улам перш за все відомий як людини яка брала участь в проектуванні водневої бомби з Едуардом Теллером на початку 50-х років. Під час II Світової Війни Станіслав Марцин

Улам і Джон фон Нейман (Neumann) працювали в Лос-Аламосі на Манхеттенському Проекті над моделюванням нейтронної дифузії в розщеплюваному матеріалі. Метод Монте-Карло він винайшов в 1946 р., коли одужуючи після хвороби, і, розкла- даючи пасьянси, задався питанням, яка вірогідність того, що пасьянс «складеться». Йому в голову прийшла ідея, що замість того, щоб використовувати звичайні для подібних завдань міркування комбінаторики, можна просто поставити «експеримент» велике число раз і, таким чином, підрахувавши число вдалих результатів, оцінити їх вірогідність. Він же запропо- нував використовувати комп'ютери для розрахунків методом Монте-Карло. Працюючи з Джоном фон Нейманом і Ніколасом Метрополісом (N. Metropolis), він розвивав алгоритми для комп'ютерних виконань, також як і досліджував засіб перстворе- ння не випадкових проблем у випадкові форми, які полегшили б їх розвязок через статистичне здійснення вибірки. Ця робота перетворила статистичне здійснення вибірки з математичної цікавості- на формальну методологію, що застосовується до широкого кола різноманітних проблем.

Поява перших електронних комп'ютерів, які могли з великою швидкістю генерувати псевдовипадкові числа, різко розширила круг завдань, для вирішення яких стохастичний підхід виявився ефективнішим, ніж інші математичні методи. Після цього відбувся великий прорив і метод Монте-Карло застосо- вувався в багатьох завданнях, проте його використання не завжди було виправдано через велику кількість обчислень, необхідних для отримання відповіді із заданою точністю. Роком народження методу Монте-Карло вважається 1949 рік, коли в світ виходить стаття Метрополіса і Улама «Метод Монте-Карло» в журналі Journal of American Statistical Association (Журнал американської статистичної звітності). Назва методу походить від назви міста в князівстві Монако, широко відомого своїми численними казино, оскільки саме рулетка є одним з найвідоміших генераторів випадкових чисел. Станіслав Улам пише в своїй автобіографії «Пригоди математика», що назва була запропонована Метро по- лісом на честь його дядька, який був азартним гравцем. Метод Монте-Карло — це сукупність формальних процедур, засобами яких відтворюються на ЕОМ будь-які випадкові фактори

(випадкові події, випадкові величини з довільним розподілом, випадкові вектори тощо). У межах цього підходу будується ймовірнісна модель, яка відповідає математичній чи фізичній задачі, і на ній реалізується випадкова вибірка. «Розігрування» вибірок за методом Монте-Карло є основним принципом імітаційного моделювання систем із стохастичними (випадковими, імовірними) елементами.

Метод Монте-Карло — це метод імітації для приблизного відтворення реальних явищ. Він об'єднує аналіз чутливості (сприйнятливості) і аналіз розподілювання ймовірностей вхідних змінних. Цей метод дає змогу побудувати модель, мінімізуючи дані, а також максимізувати значення даних, які використовуються в моделі. Побудова моделі починається з визначення функціональних залежностей у реальній системі. Після чого можна одержати кількісне рішення, використовуючи теорію ймовірності й таблиці випадкових чисел. [2, стр.70]

Метод статистичного моделювання (чи метод Монте-Карло) — це спосіб дослідження невизначених (стохастичних) економічних об'єктів і процесів, коли не повністю (до певної міри) є відомими внутрішні взаємодії в цих системах.

Цей метод полягає у модельному відтворенні процесу за допомогою стохастичної математичної моделі та обчисленні характеристик цього процесу. Одне таке відтворення можливого (випадкового) стану функціонування модельованої системи називають реалізацією (чи імітаційним прогоном). Після кожного прогону реєструють сукупність параметрів, що характеризують випадкову подію (її реалізацію). Метод ґрунтується на багатократних прогонах (випадкових реалізаціях) на підставі побудованої моделі з подальшим статистичним опрацюванням отриманих даних з метою визначення числових характеристик досліджуваного об'єкта (процесу) у вигляді статистичних оцінок його параметрів. Процес моделювання економічної системи зводиться до машинної імітації досліджуваного процесу, котрий моделюється на ЕОМ з усіма суттєвими невизначеностями, випадковостями і породженим ними ризиком.
Метод Монте-Карло широко використовується у всіх випадках імітації на ЕОМ. На сьогодні він охоплює будь-яку техніку

статистичного здійснення вибірки, яке використовується для приблизних рішень кількісних проблем.

Він приміняється:

1. Для визначення площі довільних фігур.
2. Для вибору найкращих стратегій в задачах, де присутні багато випадкових факторів.
3. Для визначення ймовірності, чи відбудеться якась подія.
4. Для побудови різних різних геометричних об’’єктів, в тому числі лабіринтів та фракталів.
5. Для моделювання поведінки складних екологічних та економічних систем.

Розв’язування задач методом статистичного моделювання полягає в слідуючому:

1. Опрацювання й побудова структурної схеми процесу, виявлення основних взаємозв’язків;
2. Формалізований опис процесу;
3. Моделювання випадкових явищ (випадкових подій, випадкових величин, випадкових функцій), що притаманні досліджуваній системі;
4. Моделювання процесу функціонування системи (на підставі використання даних, що отримані на попередньому етапі) — відтворення процесу відповідно до розробленої структурної схеми і формалізованого опису (імітаційні прогони);
5. Накопичення результатів моделювання (імітаційних прогонів), статистичне опрацювання, аналіз та інтерпретація їх.

Будь-які твердження стосовно характеристик модельованої системи повинні ґрунтуватися на результатах відповідних перевірок за допомогою методів математичної статистики.

Оскільки випадкові події й випадкові функції можуть подаватися з використанням випадкових величин, то й моделювання випадкових подій і випадкових функцій проводиться за допомогою випадкових величин.

РОЗДІЛ 2. ГЕНЕРАЦІЯ ВИПАДКОВИХ ЧИСЕЛ.

2.1. Способи отримання випадкових чисел

У програмуванні достатньо часто знаходять застосування послідовності чисел, вибраних випадковим чином з деякої множини. Як приклади завдань, в яких використовуються випадкові числа, можна привести наступні:

– тестування алгоритмів;
– імітаційне моделювання;
– деякі завдання чисельного аналізу;
– імітація користувацького введення;
– розв’язання задач автентифікації та ідентифікації та ін.

Раніше вчені,яким були потрібні для роботи випадкові числа, розкладали карти, кидали гральні кості, виймали кулі з урни тощо. Одним з найпростіших механічних приладів для отримання випадкових чисел є рулетка. В першій половині XX століття було сконструйовано спеціальні машини, які виробляли випадкові числа механічним шляхом. Зокрема, у 1955 році компанія RAND Corporation опублікувала відомі таблиці з мільйоном випадкових цифр, отриманих такою машиною.

В сучасних умовах для отримання випадкових чисел застосовують різноманітні генератори, які поділяються на дві групи- апаратні (фізичні) та програмні.

Апаратні ГВЧ є пристроями, що перетворюють в цифрову форму який-небудь параметр навколишнього середовища або фізичного процесу, тобто це спеціальні електронні приставки до ЕОМ, які утворюють випадкові числа, використовуючи фізичні явища. Параметр і процес вибираються так, щоб забезпечити хорошу «випадковість» значень при зчитуванні. Дуже часто використовуються процеси в електроніці (витоки струму, тунельний пробій діодів, цифровий шум відеокамери, шуми на мікрофонному вході звукової карти, радіоактивне випромінювання і т.п.). Для прикладу розглянемо шумлячий радіоелектронний прилад (діод, тріод) та процес радіоактивного випромінювання.

В першому випадку при роботі такого приладу рахуватимемо, скільки разів **v** за фіксований час Δt напруга перевищила заданий рівень Е0**.** Двійкове випадкове число одержуємо за допомогою співвідношення μ=v mod 2. Якщо частоти появи 0 і 1 рівні, то можна вважати, що пристрій виробляє випадкову послідовність двійкових цифр. Якщо ж частоти не рівні то вводимо яку-небудь схему стабілізації: групування чисел парами, трійками, і т.д. Зазвичай датчики містять декілька генераторів описаного типу, що працюють незалежно один від одного. Отже датчиком видається число $0, a_1, \cdots, a_m$, записане у формі m-розрядного двійкового дробу.

Сутність методу, що грунтується на радіоактивному випромінюванні, полягає у слідуючому:

1. Обирається джерело радіоактивного випромінювання з інтенсивністю λ.
2. Залежно від значення λ обирається відрізок часу Δτ.
3. За допомогою лічильника визначається кількість часточок, що їх випромінює джерело за час Δτ.
4. Застосовується схема:
   1) якщо кількість часточок парна, то $z_i = 0$;
   2) якщо кількість часточок непарна, то $z_i = 1$.

Щоб дістати *m*-розрядне випадкове двійкове число, достатньо *m* разів звернутися до лічильника радіоактивних часточок.

Формована таким чином послідовність чисел, як правило, носить абсолютно випадковий характер і не може бути відтворена наново за бажанням користувача.

Програмний генератор випадкових чисел являє собою програму, яка генерує послідовність чисел за деяким алгоритмом. Для формування чергового числа послідовності тут використовуються різні перетворення алгебри. Завдяки алгоритму, така послідовність чисел цілком детермінована (визначена), тобто принципово не може бути випадковою. Але, оскільки така числова послідовність за своїм зовнішнім виглядом та властивостями дуже нагадує випадкову, то її називають послідовністю псевдовипадкових чисел. Пристрої або алгоритми отримання

випадкових чисел називають генераторами випадкових чисел (ГВЧ) або датчиками випадкових чисел (ДВЧ).

Однією з переваг псевдовипадкових чисел є їх швидка генерація, до того ж вони не вимагають пристроїв, що запам'ятовують. Запас псевдовипадкових чисел обмежений. Як стандартні зазвичай використовують рівномірно розподілені на інтервалі [0,1] псевдовипадкові числа.

Більшість алгоритмів обчислення випадкових (псевдовипадкових) чисел, що використовуються при практичних розрахунках, засновані на рекурентних формулах першого порядку: $\xi_{k+1} = f(\xi_k)$, де ξ_o задане. Що б функція $y = f(x)$ породжувала псевдовипадкові числа, її графік повинен щільно заповнювати квадрат [(0,1)*(0,1)], тобто вона повинна бути розривною в кожній точці. Оскільки рівномірно розподілені випадкові числа повинні задовільняти статистичні вимоги (наприклад, математичне сподівання рівне 0,5, дисперсія рівна 1/12 і т.д.), то умова щільного заповнення всього квадрата є необхідною. Ще одна особливість алгоритмів типу $\xi_{k+1}=f(\xi_k)$ полягає в тому, що вони завжди породжують періодичні послідовності. Отже існують такі L і l, що $\xi_{L+i}=\xi_{l+i}$, (i=1,2…).

Існує безліч генераторів псевдовипадкових чисел. Приклади алгоритмів отримання псевдовипадкових чисел:

1. Метод середини квадратів.

Одним з перших програмних ГВЧ є метод середини квадратів, запропонований в 1946 р. Джом фон Нейманом. Цей ГВЧ формує наступний елемент послідовності на основі попереднього шляхом піднесення його до квадрату і виділення середніх цифр отриманого числа. Наприклад, ми хочемо одержати 10-значне число і попереднє число дорівнювало 5772156649. Підносимо його до квадрату і одержуємо 33317792380594909201; це означає, що наступним числом буде 7923805949. Очевидним недоліком цього методу є зациклення у випадку, якщо чергове число буде рівне нулю.

2. Лінійний конгруентний метод

Запропонований Д. Х. Лемером. Основна обчислювальна формула:

$$x[n+1] = (a\, x[n] + c) \bmod m. \qquad (2.1)$$

Алгоритм зациклюється з періодом, що не перевищує деякого m. Коефіцієнти *a, m* і *x(0)* можуть приймати довільні цілі значення, за винятком 0. Параметр *c* може бути також і 0, але в цьому випадку скорочується період.

Крім лінійних існують інші конгруентні методи. Найвідоміші з них : мультиплікативний, мішаний і адитивний. Тепер майже всі стандартні бібліотечні програми обчислення послідовності рівномірно розподілених випадкових чисел грунтуються на конгруентних методах.

3. "Mother-of-All" random number generator

Запропонований Джорджем Марсалієй (Marsaglia), професором університету Флоріди. Обчислювальна формула:

$$S = 21111111111\,x[n-4]+1429\,x[n-3]+1776\,x[n-2]+5115\,x[n-1]+c,$$
$$x[n]= S/2^{32},\ C = [S/2^{32},] . \qquad (2.2)$$

Цей алгоритм є узагальненням попереднього і позбавлений його головного недоліку – короткого періоду.Випадкове число x[i] належатиме проміжку [0, 1]. Початкові значення можна задавати довільні. Алгоритм може бути застосований в прикладних науках, але він має нижчу швидкість.

4. Mersenne Twister

Цей генератор псевдовипадкових чисел (ГПВЧ) представлений Макуто Матсумото і Такеджі Нушиміро в 1997 р. Основна ідея полягає в тому, що до початкової ітерації, яка ініціює процедуру, застосовується серія бітових операцій. Після їх виконання отримують нову послідовність, перший член якої вважається псевдовипадковим числом. Послідовність має велику розмірність (624 елементи), тому період генератора рівний (2^{19937}–1). Існує два загальних варіанти алгоритму, запропонованих авторами, найбільш поширений алгоритм MT19937. Реалізації цього алгоритму вже використовуються в стандартних бібліотеках для PHP, Python и Рубі. Алгоритм дуже швидкий через відсутність множень, але не має достатньої випадковості. Тому галузь застосування алгоритму дещо обмежена.

5. Генератори типу "Xorshift"

Одні з генераторів від Джорджа Марсалії. Знову розглядається деяка початкова послідовність, до якої застосовуються операції «Xorshift». Ці операції полягають в наступному:

$$x[i]{:}= x[i]\ xor\ (x[i]\ shl\{\ або\ shr\}\ a). \qquad (2.3)$$

Підсумкове випадкове число може бути одержано за допомогою підсумовування окремих членів послідовності, або застосування до них операції *xor*. В даний час це один з найбільш вживаних алгоритмів; послідовність, що генерується достатньо випадкова, періоди — залежно від реалізації, відсутність операцій множеня позитивно позначається на швидкості.

Є також комбінації вже представлених методів. Деякі з них реалізовані у відповідних програмних середовищах у вигляді функцій Rand().

Будь-який генератор псевдовипадкової послідовності (ГПВП) з обмеженими ресурсами рано чи пізно зациклюється. Довжина циклів ГПВП залежить від самого генератора й у середньому становить близько $2^{(n/2)}$, де *n* – це розмір внутрішнього стану в бітах, хоча лінійні-конгруентні генератори мають максимальні цикли порядку 2^n. Якщо ГПВП може сходитися до занадто коротких циклів, такий ГПВП стає передбачуваним і є непридатним.

Більшість простих арифметичних генераторів хоча й мають велику швидкість, але існує багато серйозних недоліків:

– занадто короткий період/періоди;
– послідовні значення не є незалежними;
– деякі біти “менш випадкові”, ніж інші;
– нерівномірний одномірний розподіл;

Послідовності випадкових чисел, сформовані тим чи іншим ГВЧ, повинні задовільняти ряд вимог. По-перше, числа повинні вибиратися з певної множини (найчастіше це дійсні числа в інтервалі від 0 до 1 або цілі від 0 до *N*). По-друге, послідовність повинна підкорятися певному розподілу на заданій множині (найчастіше розподіл рівномірний). Необов'язковою є вимога відтворюваності послідовності. Якщо ГВЧ дозволяє відтворити наново одного разу сформовану послідовність, відладка програм з використанням такого ГВЧ значно спрощується.

Оскільки псевдовипадкові числа не є дійсно випадковими, якість ГВЧ дуже часто оцінюється завдяки "випадковості" одержуваних чисел. У цю оцінку можуть входити різні показники, наприклад, довжина циклу (кількість ітерацій, після якого ГВЧ зациклюється), взаємозалежності між сусідніми числами (можуть виявлятися за допомогою різних методів теорії ймовірності і математичної статистики) і т.п.

Отже, якісний ГПВП має задовольняти такі вимоги:

1. Непередбачуваність;
2. Добрі статистичні властивості, псевдовипадкова послідовність за своїми статистичними властивостями не повинна істотно відрізнятися від істинної випадкової послідовності;
3. Великий період формованої послідовності: наприклад, при шифруванні для перетворення кожного елемента вхідної послідовності необхідно використовувати свій елемент псевдовипадкової гами;
4. Ефективна апаратна й програмна реалізація.

Практично всі алгоритми генерування псевдовипадкових чисел орієнтовані на конкретні машини (враховують особливості роботи процесора і способи зберігання інформації) і компілятори (враховуються особливості арифметики, реалізованої в конкретній мові програмування для конкретної машини). Тому готові генератори псевдовипадкових чисел перед використанням необхідно ретельно перевірити і набудувати їх на конкретні умови.

2.2. Генератори ПВЧ та метод статистичних випробувань Монте-Карло.

Основна проблема у методі Монте-Карло полягає в тому, щоб дістати рівномірну випадкову послідовність чисел (РВП), розподілених на відрізку [0, 1]. При побудові стохастичних

імітаційних моделей ці числа дають змогу генерувати випадкові події або випадкові величини з довільним розподілом.
Такі числа надзвичайно важливі для методу Монте-Карло. Вони дають змогу імітувати на машині ситуації зі складною стохастичною природою. Опишемо властивості цих чисел.

Випадкова величина X має рівномірний розподіл на відрізку $[a, b]$, коли її щільність розподілу ймовірностей має вигляд

$$f(x)=\begin{cases}0 & \text{при } x<a;\\ \dfrac{1}{b-a} & \text{при } a\le x\le b;\\ 0 & \text{при } x>b.\end{cases} \quad (2.4)$$

Математичне сподівання та дисперсія випадкової величини

$$m_x=\frac{a+b}{2}, \quad (2.5)$$

$$\sigma_X^2=\frac{(b-a)^2}{12}. \quad (2.6)$$

Якщо випадкова величина розподілена на відрізку [0, 1], то

$$f(x)=\begin{cases}0 & \text{при } x<0;\\ 1 & \text{при } 0\le x\le 1;\\ 0 & \text{при } x>1;\end{cases} \quad (2.7)$$

$$m_x=0{,}5;$$

$$\sigma_X^2=\frac{1}{12}.$$

Рівномірно розподілену на відрізку [0, 1] випадкову величину позначимо ξ. Для неї характерна властивість: імовірність того, що значення цієї випадкової величини потраплять на деякий інтервал з межами $0\le\alpha\le\beta\le 1$, дорівнює довжині цього інтервалу:

$$P(\alpha\le\xi\le\beta)=\int_\alpha^\beta d\xi=\beta-\alpha. \quad (2.8)$$

Ця властивість часто використовується в методі Монте-Карло як необхідна і достатня умова того, що деяка випадкова величина має розподіл (2.7).

Принципова можливість генерувати послідовні реалізації випадкової величини ξ випливає з такого перетворення:

$$\xi = z_1 2^{-1} + z_2 2^{-2} + \ldots + z_i 2^{-i} + \ldots, \qquad (2.9)$$

де z_i — реалізація випадкової величини Z, що набуває лише двох значень — 0 і 1 — з однаковою ймовірністю 0,5.

Покажемо, що отримувана з допомогою перетворення (2.9) випадкова величина ξ має властивість (2.8). Наприклад,

$$P\left(\frac{1}{4} \le \xi \le \frac{1}{2}\right) = P\left(z_1 = 0 \cap z_2 = 1\right) = P\left(z_1 = 0\right) \cdot P\left(z_2 = 1\right) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}.$$

Випадкова величина ξ, рівномірно розподілена на відрізку [0, 1], може мати нескінченну кількість реалізацій. Проте при машинному використанні методу Монте-Карло на ЕОМ можна утворити лише 2^n випадкових чисел, що не збігаються одне з одним (n — кількість двійкових розрядів машинної пам'яті). Тому рівномірна випадкова послідовність чисел (скорочено РВП [0, 1]), використана при машинних розрахунках, фактично є реалізацією дискретної випадкової величини, розподіл якої називається квазірівномірним (від лат. *quasi* — майже, ніби, неначе).

Від сукупності чисел 0, 1, 2, ..., $2^n - 1$, які можна подати з допомогою двійкових розрядів, легко перейти до можливих значень дискретної випадкової величини ξ, що має квазірівномірний розподіл на інтервалі [0, 1]:

$$x_i = \frac{i}{2^n - 1}, \qquad (i = 0, 1, 2, \ldots, 2^n - 1). \qquad (2.10)$$

В останньому виразі знаменник має вигляд $2^n - 1$, а не 2^n для того, щоб до сукупності 2^n величин x_i можна було включати як 0, так і 1, а інтервали між ними на числовій осі були однакові. Крім того, математичне сподівання величини x_i дорівнює 0,5, а при діленні на 2^n

оцінка математичного сподівання була б зсуненою $\left(\frac{1}{2}\left(1 - \frac{1}{2^n}\right)\right)$.

Імовірності, що відповідають можливим значенням x_i, мають вигляд

$$p = \frac{1}{2^n}.$$

Визначимо математичне сподівання і дисперсію дискретної квазірівномірної випадкової величини ξ:

$$m_\xi = \sum_{i=0}^{2^n - 1} \frac{i}{2^n - 1} \cdot \frac{1}{2^n} = \frac{1}{(2^n - 1)\, 2^n} \sum_{i=0}^{2^k - 1} i = \frac{1}{2};$$

$$\sigma_\xi^2 = \sum_{i=0}^{2^n - 1} \frac{1}{2^n} \left(\frac{i}{2^n - 1} - \frac{1}{2}\right)^2 = \frac{1}{2^n} \sum_{i=0}^{2^n - 1} \left(\frac{i^2}{\left(2^n - 1\right)^2} - \frac{i}{2^n - 1} + \frac{1}{4}\right) =$$

$$= \frac{2 \cdot 2^n - 1}{6\left(2^n - 1\right)} - \frac{1}{4} = \frac{1}{12} \cdot \frac{2^n + 1}{2^n - 1}.$$

Отже, математичне сподівання квазірівномірної випадкової величини збігається з математичним сподіванням РВП [0, 1], а дисперсія відрізняється лише множником $\frac{2^n + 1}{2^n - 1}$, який для великих n дуже близький до 1. Наприклад,

для $n = 10$: $\quad \frac{2^{10} + 1}{2^{10} - 1} = 1{,}002.$

Тому для $n > 10$ відмінність між дисперсіями рівномірної і квазірівномірної випадкових величин стає неістотною, а це дає підставу в імітаційному моделюванні використовувати програмно створені випадкові числа.

Згідно з центральною граничною теоремою теорії ймовірностей унаслідок додавання досить великої кількості однаково розподілених незалежних випадкових величин отримуємо випадкову величину, яка має нормальний закон розподілу. Як показали дослідження, вже внаслідок складання більш ніж десяти випадкових незалежних величин з рівномірним розподілом в інтервалі (0; 1) отримуємо випадкову величину, котру з точністю, достатньою для більшості практичних задач, можна вважати розподіленою згідно з нормальним законом.

Існують три способи дістати рівномірну випадкову послідовність чисел, розподілених на відрізку [0, 1]: табличний, програмний і фізичне генерування.

При табличному способі випадкові числа беруть із спеціальних таблиць, в яких числа отримані за допомогою фізичних чи програмних генераторів. Табличний метод з огляду на повільне введення табличних даних у пам'ять ЕОМ і необхідність використовувати значний обсяг пам'яті (таблиці записуються в оперативну пам"ять), щоб зберігати їх, для машинної імітації вважається неефективним і застосовується здебільшого для ручних розрахунків. У дослідженнях на ЕОМ він застосовується насамперед для налагодження програм або дублювання особливо важливих дослідів. Апаратні (фізичні) генератори розглядалися в попередньому розділі, використання таких генераторів вимагає наявності спеціального обладнання. В зв"язку з цим більш зручним вважається застосування програмних генераторів випадкових чисел.

У разі, коли для програмної реалізації використовуються мови моделювання, що забезпечені вмонтованими генераторами випадкових послідовностей чисел, програмістові немає потреби розробляти програми утворення таких чисел. Крім того, бібліотеки більшості ЕОМ включають спеціальні стандартні підпрограми, що їх можна використати з відповідною метою. У складі трансляторів майже всіх алгоритмічних мов є стандартні процедури (чи функції), котрі генерують, псевдовипадкові числа, що є реалізаціями послідовності випадкових чисел із рівномірним законом розподілу.

2.3. Генератор випадкових чисел в Delphi.

У мові Delphi, як і в багатьох інших мов високого рівня, існує вбудована підтримка генератора випадкових чисел. Для формування чисел використовується програмний ГВЧ, що існує в програмі в єдиному екземплярі.

У Borland Delphi використовується конгруентний ГВЧ. Довжина періоду цього ГВЧ складає 232 числа.

Для роботи з ГВЧ в мові Delphi передбачені наступні функції:

random() - генерує випадкове число;

random(x) - дозволяє генерувати випадкове ціле число в діапазоні [0, x-1];

randomize() - встановлює ГВЧ на слідуюче число;

RandomRange - генерує довільне число в межах введеного діапазону;

RandSeed - встановлює ГВЧ на слідуюче значення;

RandG - генерує нормально розподілені випадкові числа з заданим середнім значенням і середньоквадратичним відхиленням;

Delphi використовує ГПВЧ, який, кожний раз при виконанні програми повертає одну і ту ж послідовність значень (232). Щоб уникнути цієї передбачуваності використовується процедура *randomize*, яка забезпечує ініціалізацію генератора випадкових чисел. Якщо треба згенерувати визначену послідовність, то треба присвоїти системній змінній *RandSeed: LongInt* конкретне значення.

Генератор не рекомендується використовувати в процесі відладки, так як послідовність, вибрану викликом *randomize()*, складно відтворити. Крім того, генератор не рекомендується викликати дуже часто або через фіксовані проміжки часу, бо це понизить якість («випадковість») послідовностей, що генеруються.

РОЗДІЛ 3. ПРЕДСТАВЛЕННЯ ІСТИННОЇ МОДЕЛІ ЗАЛЕЖНОСТІ ВПЛИВУ ЦІНИ ТОВАРУ І ПОПИТУ НА НЬОГО.

3.1. Представлення істинної моделі.

За результатами строгого зрівноваження отримана емпірична формула залежності впливу ціни товару X і попиту на нього $У$

$$y = -4{,}717425x^3 + 33{,}731505x^2 - 85{,}78331x + 88{,}244437$$

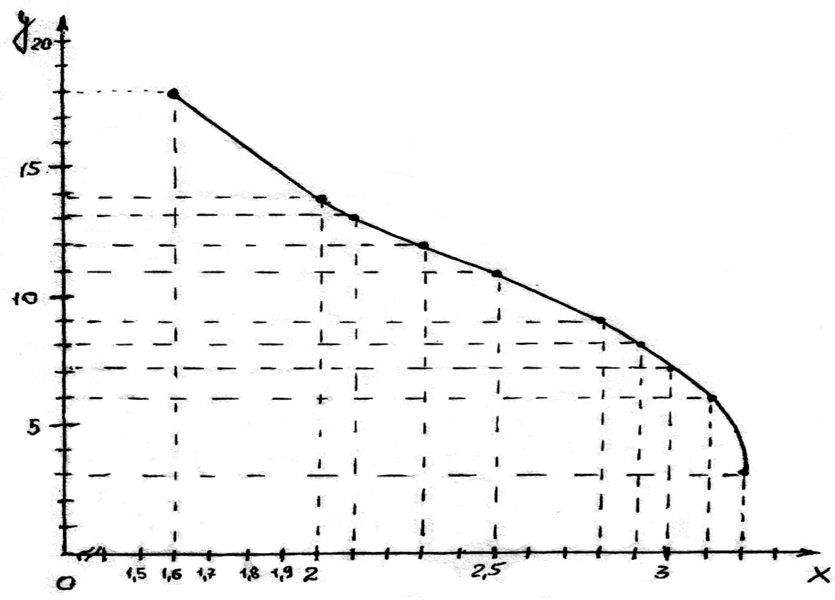
Таблиця 1. Вихідні дані істинної моделі у табличному вигляді

x	1,6	2	2,1	2,3	2,5	2,8	2,9	3	3,1	3,3
y	18,021	13,864	13,167	11,986	10,898	8,949	8,101	7,108	5,939	2,965

За даними таблиці 1 побудуємо точкову діаграму і графік (мал.2).

Побудувавши ймовірнішу модель за способом найменших квадратів (тобто знайшовши параметри (коефіцієнти) емпіричної формули) і зробивши оцінку точності її елементів, в подальшому необхідно провести дослідження точності впливу залежності ціни товару на попит на нього методом статистичних випробувань Монте- Карло. Для цього необхідно генерувати істинні похибки за допомогою генератора випадкових чисел. В даній роботі ПВЧ

генеровані за допомогою вбудованого генератора Delphi за допомогою функції *random ()*.



Мал.2. Точкова діаграма і графік істинної моделі.

3.2. Генерування істинних похибок для дослідження математичної моделі методом статистичних випробувань Монте- Карло.

Приведемо методику розрахунку випадкових чисел, які приймемо в подальшому, як істинні похибки для побудови спотвореної моделі.

1.Отримавши ряд випадкових (а точніше псевдовипадкових) чисел ξ_{cp}, розраховують середнє арифметичне генерованих псевдовипадкових чисел $\xi_{\text{ср}}$:

$$\xi_{\text{ср}} = \frac{\sum_{i=1}^{n} \xi_i}{n} \tag{3.1}$$

де n – сума випадкових чисел.

2. Розраховуються попередні значення істинних похибок Δ'_i за формулою

$$\Delta'_i = \xi_i - \xi_{\text{ср}} \tag{3.2}$$

3. Знаходять середню квадратичну похибку попередніх значень істинних похибок :

$$m_{\Delta'} = \sqrt{\frac{\sum_{i=1}^{n} \Delta'^2 i}{n}} \tag{3.3}$$

4. Знаходять коефіцієнт пропорційності K , для визначення істинних похибок необхідності точності

$$\mathrm{K} = \frac{c}{m'_{\Delta}}, \tag{3.4}$$

де c – необхідна константа.

Так, наприклад, при $m'_{\Delta} = 0{,}63254$ і необхідності побудови математичної моделі з точністю *c=0,1*, будемо мати

$$\mathrm{K}_{0,1} = \frac{0{,}1}{0{,}63254} = 0{,}15809,$$

при *c=0,05* K буде дорівнювати 0,07905.

5.Істинні похибки розраховуються за формулою

$$\Delta_i = \Delta'_i \cdot \mathrm{K} \tag{3.5}$$

6. Заключним контролем служить розрахунок середньої квадратичної похибки m_{Δ} генерованих істинних похибок Δ

$$m_{\Delta} = \sqrt{\frac{\sum_{i=1}^{n} \Delta^2}{n}} \tag{3.6}$$

і порівняння

$$m_{\Delta} = c \tag{3.7}$$

Генеровані нами похибки , розрахунок попередніх значень істинних похибок , самі істинні похибки представлені в таблиці 2.

Середня квадратична похибка попередніх істинних похибок

$$\Delta'_m = \sqrt{\frac{0{,}59769}{10}} = 0{,}24448 .$$

Коефіцієнт пропорційності

$$\mathrm{K} = \frac{0{,}1}{0{,}24448} = 0{,}40904 .$$

Для контролю вирахуємо середню квадратичну похибку і прирівняємо її до *c*.

$$m_{\Delta_i} = \sqrt{\frac{0{,}1000000}{10}} = 0{,}1$$

Отже, $m_{\Delta} = c = 0{,}1$.

Таблиця 2. Генерування псевдовипадкових чисел і розрахунок істинних похибок

№	ξ_i	ξ_{cp}	$\Delta'_i = \xi_i - \xi_{cp}$	$\Delta_i'^2$	$\Delta_i = \Delta'_i \cdot K$	Δ_i^2
1	0,01		-0,429	0,184041	-0,17547663	0,03079205
2	0,68		0,241	0,058081	0,099	0,00971758
3	0,4		-0,039	0,001521	-0,01595242	0,00025448
4	0,07		-0,369	0,136161	-0,151	0,02278121
5	0,73		0,291	0,084681	0,119029605	0,01416805
9	0,26	0,439	-0,179	0,032041	-0,073	0,00536081
7	0,71		0,271	0,073441	0,110848877	0,01228747
8	0,42		-0,019	0,000361	-0,008	0,00006040
9	0,54		0,101	0,010201	0,041312681	0,00170674
10	0,57		0,131	0,017161	0,053583774	0,00287122
Сума	4,39		-5E-16	0,59769	-2,5E-16	0,10000000

3.3. Побудова спотвореної моделі.

Визначимо $x_{спотворене}$ за формулою

$$x_{спотв.} = x_{іст} + \Delta_i \quad (3.8)$$

Дані занесено в таблицю 3.

Таблиця 3. Побудова спотвореної моделі

№	Істинна модель		Δ_i	$x_{спотв.} = x_{іст.} + \Delta_i$
	$x_{іст}$	$y_{іст}$		
1	1,6	18,021	-0,17547663	1,425
2	2	13,864	0,099	2,099
3	2,1	13,167	-0,01595242	2,084
4	2,3	11,986	-0,151	2,149
5	2,5	10,898	0,119029605	2,619
9	2,8	8,949	-0,073	2,727
7	2,9	8,101	0,110848877	3,011
8	3	7,108	-0,008	2,992
9	3,1	5,939	0,041312681	3,141
10	3,3	2,965	0,053583774	3,354
Сума	25,6	100,998	-2,5E-16	25,600

По даним спотвореної моделі виконують строге зрівноваження методом найменших квадратів і отримують ймовірніші моделі, яким роблять оцінку точності зрівноважених елементів і дають порівняльний аналіз на основі якого роблять висновок про предмет поширення даної моделі на рішення проблеми в цілому.

РОЗДІЛ 4. ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ.

4.1. Підбір параметрів способом найменших квадратів.

Метод найменших квадратів (МНК) — це математично-статистичний метод, який полягає в тому, що функція (котра може бути відомою, або заданою динамічним рядом чи таблицею експериментальних даних) для опису деякого явища апроксимується більш простою функцією (лінійною функцією, параболою, поліномами різного ступеня тощо). Апроксимуюча функція добирається таким чином, щоб середньоквадратичне відхилення (сума квадратів відхилень) фактичних рівнів функції в спостережуваних точках від вирівняних було найменшим.

На практиці часто приходиться розв”язувати таку задачу: для двох функціонально зв”язаних величин величин x та y відомі n пар відповідних значень (x_1 ; y_1), (x_2 ; y_2), …(x_n ; y_n). Потрібно в наперед заданій формулі $y = f(x, \ a_{1,} \ a_{2, \ \ldots,} \ a_m)$

визначити m параметрів $a_1, \ a_2, \ \cdots, \ a_m$ $(m<n)$ так щоб

в цю формулу найкращим чином “вкладалися” б відомі n значень
x та y.

Вважається (виходячи з принципів теорії ймовірності), що найкращими будуть такі значення $a_1, \ a_2, \ \cdots, \ a_m$, які

перетворюють в мінімум суму

$$\sum_{k=1}^{k=n}\left[f \ (x_k, \ a_1, \ a_2, \cdots, \ a_{m)} - y_k\right]^2$$

(тобто суму квадратів відхилень значень y, котрі вирахувані по формулі, від заданих, тому сам спосіб і отримав назву способу найменших квадратів).

Ця умова дає систему m рівнянь, з котрих визначаються $a_1, \ a_2, \ \cdots, \ a_m$:

$$\sum_{k=1}^{k=n}\left[f \ (x_k, \ a_1, \ a_2, \cdots, \ a_m) - y_k\right] \cdot \frac{df(x_k, \ a_1, \ a_2, \ \cdots, \ a_m)}{da_j} = 0, \qquad (4.1)$$

де $(j = 1, \ 2, \ \cdots, \ m)$.

В випадку $y = a_0 x^m + a_1 x^{m-1} + \cdots + a_m$ $(m+1$

параметрів $a_0, \ a_1, \ \cdots, \ a_m; \ n > m+1)$, система (4.1) приймає

вигляд:

$$a_0 \cdot \sum_{k=1}^{k=n} x_k^m \ + \ a_1 \cdot \sum_{k=1}^{k=n} x_k^{m-1} \ + \cdots + \ n \cdot a_m = \sum_{k=1}^{k=n} y_k,$$

$$a_0 \cdot \sum_{k=1}^{k=n} x_k^{m+1} \ + \ a_1 \cdot \sum_{k=1}^{k=n} x_k^m \ + \cdots + \ a_m \cdot \sum_{k=1}^{k=n} x_k = \sum_{k=1}^{k=n} x_k . y_k,$$

$$a_0 \cdot \sum_{k=1}^{k=n} x_k^{m+2} \ + \ a_1 \cdot \sum_{k=1}^{k=n} x_k^{m+1} \ + \cdots + \ a_m \cdot \sum_{k=1}^{k=n} x_k^2 = \sum_{k=1}^{k=n} x_k^2 . y_k, \qquad (4.2)$$

…………………………………………………………………

$$a_0 \cdot \sum_{k=1}^{k=n} x_k^{2m} \ + \ a_1 \cdot \sum_{k=1}^{k=n} x_k^{2m-1} \ + \cdots + \ a_m \cdot \sum_{k=1}^{k=n} x_k^m = \sum_{k=1}^{k=n} x_k^m . y_k$$

Для визначення коефіцієнтів системи (4.2) зручно складати допоміжну таблицю, де в останньому рядку записуються суми елементів кожного стовбчика які і є коефіцієнтами системи (4.2).

4.2. Представлення системи нормальних рівнянь.

У результаті проведеного експерименту ми маємо ряд результатів x_i , y_i функціональну залежність між якими будемо шукати за допомогою поліному степені K, де коефіцієнти a_i являються невідомими. Ввівши в системі (4.2) позначення суми відповідного елемента знаком [ ] система нормальних рівнянь для поліному третього порядку виду

$$y = ax^3 + вx^2 + cx + d \tag{4.3}$$

буде

$$\begin{aligned} dn + c[x] + в[x^2] + a[x^3] - [y] &= 0, \\ d[x] + c[x^2] + в[x^3] + a[x^4] - [xy] &= 0, \\ d[x^2] + c[x^3] + в[x^4] + a[x^5] - [x^2 y] &= 0, \\ d[x^3] + c[x^4] + в[x^5] + a[x^6] - [x^3 y] &= 0. \end{aligned} \tag{4.4}$$

або

$$\begin{aligned} a[x^6] + в[x^5] + c[x^4] + d[x^3] - [x^3 y] &= 0, \\ a[x^5] + в[x^4] + c[x^3] + d[x^2] - [x^2 y] &= 0, \\ a[x^4] + в[x^3] + c[x^2] + d[x] - [xy] &= 0, \\ a[x^3] + в[x^2] + c[x] + dn - [y] &= 0. \end{aligned} \tag{4.5}$$

В подальшому будемо розв”язувати систему лінійних нормальних рівнянь (4.4) або (4.5) одним із відомих в математиці способів.

4.3. Встановлення коефіцієнтів нормальних рівнянь.

Приведемо розрахункову таблицю, на основі якої отримують коефіцієнти нормальних рівнянь.

Таблиця 4. Розрахунок коефіцієнтів нормальних рівнянь.

№	$x_{сноте.}$	x^0	x^2	x^3	x^4
1	1,425	1	2,029	2,89073800	4,118
2	2,099	1	4,404	9,24219681	19,395
3	2,084	1	4,343	9,05154863	18,864
4	2,149	1	4,618	9,9254222	21,330
5	2,619	1	6,859	17,9647519	47,050
9	2,727	1	7,435	20,2745621	55,284
7	3,011	1	9,065	27,2939802	82,178
8	2,992	1	8,953	26,7907074	80,164
9	3,141	1	9,868	30,9979878	97,374
10	3,354	1	11,247	37,7161608	126,484
Сума	25,600	10	68,823	192,148056	552,243

Продовження таблиці 4.

№	x^5	x^6	xy	x^2y	x^3y
1	5,866	8,356	25,6713356	36,56941732	52,09398943
2	40,703	85,418	29,0946824	61,05745407	128,1338166
3	39,313	81,931	27,4406545	57,1876295	119,1817408
4	45,840	98,514	25,7586997	55,35713419	118,9661101
5	123,226	322,732	28,5421846	74,75282658	195,7798659
9	150,748	411,058	24,4019764	66,53888158	181,4370563
7	247,426	744,961	24,3908867	73,43727397	221,1085338
8	239,869	717,742	21,2687588	63,64098218	190,4283484
9	305,883	960,875	18,656256	58,60513359	184,0970493
10	424,176	1422,509	9,94337589	33,34594405	111,8284169
Сума	1623,050	4854,097	235,169	580,493	1503,055

Таким чином, на основі проведених розрахунків нами отримана наступна система нормальних рівнянь

$$\begin{aligned}
&10d+25{,}6c+68{,}823b+192{,}148a-100{,}998=0,\\
&25{,}6d+68{,}823c+192{,}148b+552{,}243a-235{,}169=0,\\
&68{,}823d+192{,}148c+552{,}243b+1623{,}050a-580{,}493=0,\\
&192{,}148d+552{,}243c+1623{,}050b+4854{,}097a-1503{,}055=0.
\end{aligned} \qquad (4.6)$$

$$\begin{aligned}
&4854{,}097a+1623{,}050b+552{,}243c+192{,}148d-1503{,}055=0,\\
&1623{,}050a+552{,}243b+192{,}148c+68{,}823d-580{,}493=0,\\
&552{,}243a+192{,}148b+68{,}823c+25{,}6d-235{,}169=0,\\
&192{,}148a+68{,}823b+25{,}6c+10d-100{,}998=0.
\end{aligned} \qquad (4.7)$$

4.4. Рішення системи лінійних рівнянь способом Крамера

Нехай, маємо систему лінійних рівнянь

$$\begin{aligned}
&a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n=b_1,\\
&a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n=b_2,\\
&\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\quad\ldots\\
&a_{n1}x_1+a_{n2}x_2+\ldots+a_{nn}x_n=b_n.
\end{aligned} \qquad (4.8)$$

Система рівнянь називається крамерівською, якщо число рівнянь співпадає з числом невідомих і визначник системи відмінний від нуля. Для того, щоб із цієї системи визначити невідомі x_i, складемо із коефіцієнтів при невідомих визначник Δ, який називається визначником системи рівнянь (4.8)

$$\Delta=\begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1n}\\ a_{21} & a_{22} & \ldots & a_{2n}\\ \ldots & \ldots & \ldots & \ldots\\ a_{n1} & a_{n2} & \ldots & a_{nn}\end{vmatrix}. \qquad (4.9)$$

Помножимо ліву і праву частини рівності (4.9) на x_i. В лівій частині будемо мати $\Delta \cdot X_i$, в правій же частині введемо у всі члени *i*-го стовпчика визначника a_{ki} множник x_i

$$\Delta x_i = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1i}x_i & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2i}x_i & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{ni}x_i & \dots & a_{nn} \end{vmatrix}. \qquad (4.10)$$

Після до *i*-го стовпчика визначника (4.10) додамо всі інші стовпчики, помножені відповідно на $x_1, x_2, \dots, x_n$. Величина визначника від цього не зміниться. Тоді *i*-ий стовпчик представить собою ліву частину системи рівнянь (4.8).

Замінимо його вільними членами цієї системи і позначимо через Δ_i:

$$\Delta_i = \begin{vmatrix} a_{11} & a_{12} & \dots & b_1 & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & b_2 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & b_n & \dots & a_{nn} \end{vmatrix}. \qquad (4.11)$$

Звідки

$$x_i = \frac{\Delta_i}{\Delta} = \frac{\begin{vmatrix} a_{11} & a_{12} & \dots & b_1 & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & b_2 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & b_n & \dots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{vmatrix}}. \qquad (4.12)$$

Формула (4.12) дає можливість визначити кожне невідоме системи лінійних рівнянь (4.8). Тобто будь- яка крамерівська система рівнянь має єдиний розв"язок ($x_1, x_2, \dots, x_n$), який дають формули

$$x_i = \frac{\Delta_i}{\Delta}, \qquad (4.13)$$

де Δ_i-визначник матриці, отриманий із матриці Δ заміною *i*-того стовпчика на стовпчик вільних членів системи (4.8). Формули (4.13) називають формулами Крамера.

Для системи чотирьох лінійних рівнянь

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3, \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= b_4, \end{aligned} \qquad (4.14)$$

якщо визначник системи Δ не дорівнює нулю

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} \neq 0, \qquad (4.15)$$

то система визначена і по Крамеру її невідомі виражаються формулами

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} & a_{13} & a_{14} \\ b_2 & a_{22} & a_{23} & a_{24} \\ b_3 & a_{32} & a_{33} & a_{34} \\ b_4 & a_{42} & a_{43} & a_{44} \end{vmatrix}}{\Delta}, \tag{4.16}$$

$$x_2 = \frac{\begin{vmatrix} a_{11} & b_1 & a_{13} & a_{14} \\ a_{21} & b_3 & a_{23} & a_{24} \\ a_{31} & b_3 & a_{33} & a_{34} \\ a_{41} & b_4 & a_{43} & a_{44} \end{vmatrix}}{\Delta}, \tag{4.17}$$

$$x_3 = \frac{\begin{vmatrix} a_{11} & a_{12} & b_1 & a_{14} \\ a_{21} & a_{22} & b_2 & a_{24} \\ a_{31} & a_{32} & b_3 & a_{34} \\ a_{41} & a_{42} & b_4 & a_{44} \end{vmatrix}}{\Delta}, \tag{4.18}$$

$$x_4 = \frac{\begin{vmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \\ a_{41} & a_{42} & a_{43} & b_4 \end{vmatrix}}{\Delta} \quad . \tag{4.19}$$

Як бачимо, що

$$\Delta_{x_1} = \begin{vmatrix} b_1 & a_{12} & a_{13} & a_{14} \\ b_2 & a_{22} & a_{23} & a_{24} \\ b_3 & a_{32} & a_{33} & a_{34} \\ b_4 & a_{42} & a_{43} & a_{44} \end{vmatrix} \quad , \tag{4.20}$$

$$\Delta_{x_2} = \begin{vmatrix} a_{11} & b_1 & a_{13} & a_{14} \\ a_{21} & b_3 & a_{23} & a_{24} \\ a_{31} & b_3 & a_{33} & a_{34} \\ a_{41} & b_4 & a_{43} & a_{44} \end{vmatrix}, \tag{4.21}$$

$$\Delta_{x_3} = \begin{vmatrix} a_{11} & a_{12} & b_1 & a_{14} \\ a_{21} & a_{22} & b_2 & a_{24} \\ a_{31} & a_{32} & b_3 & a_{34} \\ a_{41} & a_{42} & b_4 & a_{44} \end{vmatrix}, \tag{4.22}$$

$$\Delta_{x_4} = \begin{vmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \\ a_{41} & a_{42} & a_{43} & b_4 \end{vmatrix}. \tag{4.23}$$

Приведемо формулу знаходження визначника четвертого порядку

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} = a_{11}(a_{22}a_{33}a_{44} + a_{23}a_{34}a_{42} +$$

$$+ a_{24}a_{32}a_{43} - a_{24}a_{33}a_{42} - a_{23}a_{32}a_{44} - a_{22}a_{34}a_{43}) -$$

$$- a_{12}(a_{21}a_{33}a_{44} + a_{23}a_{34}a_{41} + a_{24}a_{31}a_{43} - a_{24}a_{33}a_{41} - \quad (4.24)$$

$$- a_{23}a_{31}a_{44} - a_{21}a_{34}a_{43}) + a_{13}(a_{21}a_{32}a_{44} + a_{22}a_{34}a_{41} +$$

$$+ a_{24}a_{31}a_{42} - a_{24}a_{32}a_{41} - a_{22}a_{31}a_{44} - a_{21}a_{34}a_{42}) -$$

$$- a_{14}(a_{21}a_{32}a_{43} + a_{22}a_{33}a_{41} + a_{23}a_{31}a_{42} - a_{23}a_{32}a_{41} -$$

$$- a_{22}a_{31}a_{43} - a_{21}a_{33}a_{42}).$$

І в нашому випадку

$$\Delta = \begin{vmatrix} x^6 & x^5 & x^4 & x^3 \\ x^5 & x^4 & x^3 & x^2 \\ x^4 & x^3 & x^2 & x \\ x^3 & x^2 & x & n \end{vmatrix} = \begin{vmatrix} 4854{,}097 & 1623{,}050 & 552{,}243 & 192{,}148 \\ 1623{,}050 & 552{,}243 & 192{,}148 & 68{,}823 \\ 552{,}243 & 192{,}148 & 68{,}832 & 25{,}6 \\ 192{,}148 & 68{,}823 & 25{,}6 & 10 \end{vmatrix} =$$

$$= 11{,}2811$$

$$\Delta_{x_1} = \begin{vmatrix} x^3y & x^5 & x^4 & x^3 \\ x^2y & x^4 & x^3 & x^2 \\ xy & x^3 & x^2 & x \\ y & x^2 & x & n \end{vmatrix} = \begin{vmatrix} 1503{,}055 & 1623{,}050 & 552{,}243 & 192{,}148 \\ 580{,}493 & 552{,}243 & 192{,}148 & 68{,}826 \\ 235{,}169 & 192{,}148 & 68{,}823 & 25{,}6 \\ 100{,}988 & 68{,}823 & 25{,}6 & 10 \end{vmatrix} =$$

$$= -32{,}8364$$

тоді невідомий коефіцієнт a при x^3 буде

$$a = x_1 = \frac{\Delta_{x_1}}{\Delta} = \frac{-32{,}8364}{11{,}2811} = -2{,}9107;$$

$$\Delta_{x_2} = \begin{vmatrix} x^6 & x^3y & x^4 & x^3 \\ x^5 & x^2y & x^3 & x^2 \\ x^4 & xy & x^2 & x \\ x^3 & y & x & n \end{vmatrix} = \begin{vmatrix} 4857{,}097 & 1503{,}055 & 552{,}243 & 192{,}148 \\ 1623{,}050 & 580{,}493 & 192{,}148 & 68{,}826 \\ 552{,}243 & 235{,}169 & 68{,}823 & 25{,}6 \\ 192{,}148 & 100{,}988 & 25{,}6 & 10 \end{vmatrix} =$$

$$= 222{,}2584$$

коефіцієнт b при x^2 буде

$$b = x_2 = \frac{\Delta_{x_2}}{\Delta} = \frac{222{,}2584}{11{,}2811} = 19{,}7019;$$

$$\Delta_{x_3} = \begin{vmatrix} x^6 & x^5 & x^3y & x^3 \\ x^5 & x^4 & x^2y & x^2 \\ x^4 & x^3 & xy & x \\ x^3 & x^2 & y & n \end{vmatrix} = \begin{vmatrix} 4857{,}097 & 1623{,}050 & 1503{,}055 & 192{,}148 \\ 1623{,}050 & 552{,}243 & 580{,}493 & 68{,}826 \\ 552{,}243 & 192{,}148 & 235{,}169 & 25{,}6 \\ 192{,}148 & 68{,}823 & 100{,}988 & 10 \end{vmatrix} =$$

$$= -556{,}7968$$

і невідомий коефіцієнт c при x буде

$$c = x_3 = \frac{\Delta_{x_3}}{\Delta} = \frac{-556{,}7968}{11{,}2811} = -49{,}3567;$$

коефіцієнт d буде

$$\Delta_{x_4} = \begin{vmatrix} x^6 & x^5 & x^4 & x^3y \\ x^5 & x^4 & x^3 & x^2y \\ x^4 & x^3 & x^2 & xy \\ x^3 & x^2 & x & y \end{vmatrix} = \begin{vmatrix} 4857{,}097 & 1623{,}050 & 552{,}243 & 1503{,}055 \\ 1623{,}050 & 552{,}243 & 192{,}148 & 580{,}493 \\ 552{,}243 & 192{,}148 & 68{,}823 & 235{,}169 \\ 192{,}148 & 68{,}823 & 25{,}6 & 100{,}988 \end{vmatrix} =$$

$$= -640{,}6381$$

$$d = \frac{\Delta_{x_4}}{\Delta} = \frac{-640{,}6381}{11{,}2811} = 56{,}7887.$$

Таким чином, на основі проведених досліджень, математична модель впливу залежності ціни товару x_i на попит y_i виражається формулою

$$y = -2{,}9107\,x^3 + 19{,}7019\,x^2 - 49{,}3567\,x + 56{,}7887 \qquad (4.25)$$

РОЗДІЛ 5. ОЦІНКА ТОЧНОСТІ РЕЗУЛЬТАТІВ ЗРІВНОВАЖЕННЯ.

5.1. Контроль зрівноваження.

Підставляючи отримані значення коефіцієнтів $a, в, c, d$ у формулу (4.7), отримаємо наступні результати (таблиця 5).

Таблиця 5. Коефіцієнти нормальних рівнянь

і контроль зрівноваження.

	$x^3]$	$x^2]$	$x]$	$x^0]$	y	Контроль
$[x^3$	4854,097	1623,050	552,243	192,148	1503,055	1503,055
$[x^2$	1623,050	552,243	192,148	68,823	580,493	580,493
$[x$	552,243	192,148	68,823	25,6	235,169	235,169
$[x^0$	192,148	68,823	25,6	10	100,988	100,988
	-2,9107	19,7019	-49,3567	56,7887		
	a	b	c	d		

КОНТРОЛЬ ЗРІВНОВАЖЕННЯ:

$[Y\,Y]-a[Y\,X\text{^}3]-b[Y\,X\text{^}2]-c[Y\,X]-d[Y]$

= **2,888854**

[VV]= **2,888854**

0,000000

5.2. Оцінка точності параметрів, отриманих із рішення системи нормальних рівнянь.

Середні квадратичні похибки x_1, x_2, x_3, x_4 розраховуються за формулами:

$$m_{x_1} = m\sqrt{\frac{A_{11}}{\Delta}}, \tag{5.1}$$

$$m_{x_2} = m\sqrt{\frac{A_{22}}{\Delta}}, \tag{5.2}$$

$$m_{x_3} = m\sqrt{\frac{A_{33}}{\Delta}}, \tag{5.3}$$

$$m_{x_4} = m\sqrt{\frac{A_{44}}{\Delta}}, \quad . \tag{5.4}$$

де $m_{x_1}, m_{x_2}, m_{x_3}, m_{x_4}$ − середні квадратичні похибки визначаємих невідомих x_1, x_2, x_3, x_4, m – середня квадратична похибка одиниці ваги, яка розраховується за формулою

$$m = \sqrt{\frac{[VV]}{n-K}} \tag{5.5}$$

У формулі (5.5) n - число парних факторів X і $У$, К - число коефіцієнтів, що визначаються. В нашому випадку $n=10; K=4$. V - різниця між вихідним значенням y_i і вирахуваним значенням y' за отриманою нами, формулою (4.25)

$$V_i = y_i - y_i' \qquad (5.6)$$

$A_{11}, A_{22}, A_{33}, A_{44} -$ алгебраїчні доповнення першого, другого, третього і четвертого діагональних елементів

$$A_{11} = \begin{vmatrix} a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} \end{vmatrix}, \qquad (5.7)$$

$$A_{22} = \begin{vmatrix} a_{11} & a_{13} & a_{14} \\ a_{31} & a_{33} & a_{34} \\ a_{41} & a_{43} & a_{44} \end{vmatrix}, \qquad (5.8)$$

$$A_{33} = \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{41} & a_{42} & a_{44} \end{vmatrix}, \qquad (5.9)$$

$$A_{44} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}, \qquad (5.10)$$

де
$$\Delta = a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13} + a_{14}A_{14}. \qquad (5.11)$$

Приведемо формулу розкриття визначника третього порядку

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}(a_{22}a_{33} - a_{23}a_{32}) + \qquad (5.12)$$
$$+ a_{12}(a_{23}a_{31} - a_{21}a_{33}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}).$$

І в нашому випадку отримаємо

$$A_{11} = \begin{vmatrix} x^4 & x^3 & x^2 \\ x^3 & x^2 & x \\ x^2 & x & n \end{vmatrix} = \begin{vmatrix} 552{,}243 & 192{,}148 & 68{,}823 \\ 192{,}148 & 68{,}823 & 25{,}6 \\ 68{,}823 & 25{,}6 & 10 \end{vmatrix} = 35{,}0623,$$

величина оберненої ваги

$$\frac{1}{P_a} = \frac{A_{11}}{\Delta} = \frac{35{,}0623}{11{,}2811} = 3{,}1578, \quad \text{звідси}$$

$$\sqrt{\frac{1}{P_a}} = 1{,}7630 .$$

Алгебраїчне доповнення другого діагонального елемента:

$$A_{22} = \begin{vmatrix} x^6 & x^4 & x^3 \\ x^4 & x^2 & x \\ x^3 & x & n \end{vmatrix} = \begin{vmatrix} 4854{,}097 & 552{,}243 & 192{,}148 \\ 552{,}243 & 68{,}823 & 25{,}6 \\ 192{,}148 & 25{,}6 & 10 \end{vmatrix} = 1778{,}050,$$

величина оберненої ваги

$$\frac{1}{P_b} = \frac{A_{22}}{\Delta} = \frac{1778{,}050}{11{,}2811} = 157{,}6132 , \qquad \sqrt{\frac{1}{P_b}} = 12{,}5544 .$$

Алгебраїчне доповнення третього діагонального елемента:

$$A_{33} = \begin{vmatrix} x^6 & x^5 & x^3 \\ x^5 & x^4 & x^2 \\ x^3 & x^2 & n \end{vmatrix} = \begin{vmatrix} 4854{,}097 & 1623{,}050 & 192{,}148 \\ 1623{,}050 & 552{,}243 & 68{,}823 \\ 192{,}148 & 68{,}823 & 10 \end{vmatrix} = 9364{,}373,$$

величина оберненої ваги

$$\frac{1}{P_c} = \frac{A_{33}}{\Delta} = \frac{9364{,}373}{11{,}2811} = 830{,}0940, \qquad \sqrt{\frac{1}{P_c}} = 28{,}8114.$$

Алгебраїчне доповнення четвертого діагонального елемента:

$$A_{44} = \begin{vmatrix} x^6 & x^5 & x^4 \\ x^5 & x^4 & x^3 \\ x^4 & x^3 & x^2 \end{vmatrix} = \begin{vmatrix} 4854{,}097 & 1623{,}050 & 552{,}243 \\ 1623{,}050 & 552{,}243 & 192{,}148 \\ 552{,}243 & 192{,}148 & 68{,}823 \end{vmatrix} = 5083{,}582$$

величина оберненої ваги

$$\frac{1}{P_d} = \frac{A_{44}}{\Delta} = \frac{5083{,}582}{11{,}2811} = 450{,}6282, \qquad \sqrt{\frac{1}{P_d}} = 21{,}228.$$

Підставляючи у виведену нами, формулу (4.25) значення X спотвореної моделі отримаємо розрахункові значення y', які будуть дещо відрізнятися від вихідних значень Y.

Таблиця 6. Порівняльний аналіз результатів строгого зрівноваження

№	$x_{спотв.}$	$y_{іст.}$	$y'_{зрівноваж}$	$V = y_i - y'_i$	V^2
1	1,425	18,021	18,04512102	-0,02412102	0,000581824
2	2,099	13,864	13,07580304	0,788196958	0,621254445
3	2,084	13,167	13,15052466	0,016475342	0,000271437
4	2,149	11,986	12,82034481	-0,834344808	0,696131259
5	2,619	10,898	10,37264394	0,525356057	0,275998987
9	2,727	8,949	9,67985142	-0,73085142	0,534143799
7	3,011	8,101	7,338990212	0,762009788	0,580658918
8	2,992	7,108	7,52064966	-0,41264966	0,170279742
9	3,141	5,939	5,931705161	0,007294839	5,32147E-05
10	3,354	2,965	3,062366077	-0,097366077	0,009480153
Сума	25,600	100,998	101,00	0,000	2,889

Тоді, середня квадратична похибка одиниці ваги буде

$$m = \sqrt{\frac{[VV]}{n - \mathrm{K}}} = \sqrt{\frac{2{,}889}{6}} = 0{,}6934$$

Середня квадратична похибка визначення коефіцієнта a

$$m_a = m\sqrt{\frac{1}{P_a}} = 1{,}2233$$

Середня квадратична похибка визначення коефіцієнта b

$$m_b = m\sqrt{\frac{1}{P_b}} = 8{,}7113$$

Середня квадратична похибка визначення коефіцієнта c

$$m_c = m\sqrt{\frac{1}{P_c}} = 19{,}9917$$

Середня квадратична похибка визначення коефіцієнта d

$$m_d = m\sqrt{\frac{1}{P_d}} = 14{,}7298.$$

Проведемо контрольні розрахунки функцією ЛИНЕЙН в табличному процесорі Exel, яка обраховує статистику для ряду з приміненням МНК (табл.7).

Таблиця 7. Контрольні розрахунки

	Контрольні розрахунки функцією ЛИНЕЙН					
	A	B	C	D	E	F
	c	b	a	d	**F(0,05;4;6)**	4,533677
1	-49,35669	19,7018762	-2,910746	56,78872	c,b,a,d	
2	19,99177	8,71131755	1,2233493	14,7298	стандарт S	ai=S√dii
3	0,983302	0,69388445	#Н/Д	#Н/Д	R^2	μ=m
4	117,7743	6	#Н/Д	#Н/Д	Fкритерій	n-m-1
5	170,1164	2,88885378	#Н/Д	#Н/Д	[(Y'-Ycp)^2]	[VV]
6	-2,468851	2,26164138	-2,379326	3,855364	**t(0,05;6)**	2,4469118

В першому рядку приведені значення коефіцієнтів *a, b, c, d* , в другому– значення середніх квадратичних похибок для коефіцієнтів. Значення коефіцієнтів та похибок повністю співпадають зі значеннями обрахованими програмою, що говорить про те, що програма працює вірно.

Крім того, абсолютна автентичність середньої квадратичної похибки одиниці ваги (стовпчик В рядок 3), рівна 0,6939 тисяч гривень, забезпечила автентичність коефіцієнтів проведеним нами розрахункам.

В рядку 6 приведена статистична значущість коефіцієнтів моделі. Як бачимо:

t(c)=2,4688509>2,4469136 (знак мінус до уваги не приймається).

t(b)=2,261641378<2,4469136; ?

t(a)=2,3793257<2,4469136; ?

t(d)=3,85536385>2,4469136.

Таким чином, встановлені нами в даній дипломній роботі коефіцієнти c та d статистично значимі на рівні 95% t – розподілу Стьюдента. Коефіцієнти a та b статистично незначимы. Це і буде однією із характеристик адекватності побудованої нами математичної моделі емпіричним даним результатів експерименту.

Отриманий параметр F-розподілу Фішера (стовпчик А, рядок 4):

F=117,77431>4,53368898

повністю підтверджує з надійністю 95% адекватність побудованої нами в даній дипломній роботі математичній моделі залежності попиту на товар від його ціни

$$y = -2{,}9107x^3 + 19{,}7017x^2 - 49{,}356x + 56{,}788$$

емпіричним даним проведеного нами експерименту.

Коефіцієнт детермінації R^2=0,9833 (стовпчик А рядок 3) говорить про вдалий вибір рівняння математичної моделі впливу ціни товару на попит для прогнозування значень Y.

Помноживши матрицю коефіцієнтів початкових рівнянь φ (таблиця 8)

Таблиця 8

Матриця коефіцієнтів початкових рівнянь φ			
1	1,425	2,029	2,89073800
1	2,099	4,404	9,24219681
1	2,084	4,343	9,05154863
1	2,149	4,618	9,925422
1	2,619	6,859	17,964752
1	2,727	7,435	20,274562
1	3,011	9,065	27,293980
1	2,992	8,953	26,790707
1	3,141	9,868	30,997988
1	3,354	11,247	37,716161

На обернену матрицю Q (таблиця 9)

Таблиця 9

Обернена матриця Q			
3,108	-22,06809302	50,10534733	-36,1170009
-22,068093	157,6134882	-360,2834966	261,6212551
50,10534733	-360,2834966	830,0955128	-608,240647
-36,1170009	261,6212551	-608,2406474	450,6290408

Отримаємо допоміжну матрицю Q’ (табл. 10)

Таблиця 10

Допоміжна матриця Q			φ*Q=Q'
-0,5374541	4,4357427	-12,020107	10,67057177
0,57221844	-4,28501814	10,1639716	-7,424180687
0,59275584	-4,42164889	10,4460346	-7,600554384
0,4930216	-3,7515604	9,0426365	-6,703211214
-0,4213225	2,70102872	-5,3641712	3,338697822
-0,5544915	3,69590205	-7,7180297	5,076405191
-0,4706868	3,33551964	-7,420047	5,183529196
-0,501226	3,53232623	-7,8208101	5,441809915
-0,133	1,097	-2,708	2,038716323
0,960	-6,339	13,398	-9,021783935

Перемноживши порядково елементи матриці Q на транспоновані рядки матриці φ (таблиця 11)

Таблиця 11

Транспонована матриця									
2,891	9,242	9,052	9,92542	17,965	20,275	27,294	26,791	30,998	37,716
2,029	4,404	4,343	4,61848	6,8593	7,4353	9,0652	8,9534	9,8678	11,247
1,425	2,099	2,084	2,14907	2,619	2,7268	3,0108	2,9922	3,1413	3,3536
1,000	1,000	1,000	1	1	1	1	1	1	1

отримаємо вектор обернених ваг зрівноваженої функції

1/P=Q*φ™
0,995314305
0,322916786
0,330491433
0,296937893
0,248031022
0,269243624
0,233160656
0,238398666
0,233240673
0,832264941

Квадратний корінь із обернених ваг буде

√(1/P)
0,997654401
0,568257676
0,574883843
0,544920079
0,49802713
0,518886909
0,482867121
0,488260858
0,48294997
0,912285559

і середні квадратичні похибки зрівноваженої функції m_{φ}=

mφ
0,692256875
0,394305165
0,398902959
0,378111569
0,345573
0,360047557
0,335053986
0,338796617
0,335111474
0,633020763

РОЗДІЛ 6. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

6.1. Засоби розробки програмного продукту

Бурхливий розвиток обчислювальної техніки, потреба в ефективних засобах розробки програмного забезпечення привели до появи систем програмування, орієнтованих на так звану “швидку розробку”, серед яких можна виділити Borland Delphi. В основі систем швидкої розробки (RAD- систем, Rapid Application Development- середовище швидкої розробки додатків) лежить технологія візуального проектування і подійного програмування, сутність яких заключається в тому, що середовище розробки бере на себе велику частину генерації кода програми, залишаючи програмісту роботу по конструюванню діалогових вікон та функцій обробки подій. Delphi являє собою достатньо могутню систему візуального об”єктно-орієнтованого програмування, в якій в якості мови програмування використовується Object Pascal. Використовуючи Delphi можна створювати віконні інтерфейси, що будуть задовольняти стандарти Windows, причому швидко і легко. Спектр галузей в яких можливе використання Delphi достатньо широкий: інженерні, офісні, торгові та ін. Використовуючи Delphi, можна розробляти бібліотеки .DLL компонентів, форм, функцій, працювати з віддаленими та локальними базами даних будь- яких типів. Поширене також використання Delphi при розробці різних компонентів програмного забезпечення. Прикладом категорій програмних продуктів можуть бути: утиліти, інформаційне забезпечення в Інтернеті, веб-проектування, музичне оформлення, інженерне програмне забезпечення, створення баз даних і ін.

Саме тому для розробки даного програмного продукту обрано технологію роботи в середовищі Delphi. Простота мови та ефек- тивність (невеликі розміри і висока продуктивність) створюваних за його допомоги програм зробили Delphi незамінним засобом розробки додатків. Використовуючи засоби Delphi, програміст має можливість встановлювати зв”язок між особистими додатка- ми та такими продуктами Microsoft, як World, Exel та іншими, використовуючи їх можливості, створювати могутні

системи допомоги. Робота з Delphi здійснюється на таких платформах, як GNU/Linux, Mac OS X і Windows. Багато в чому Delphi став основою для створення такої мови програмування як C#.

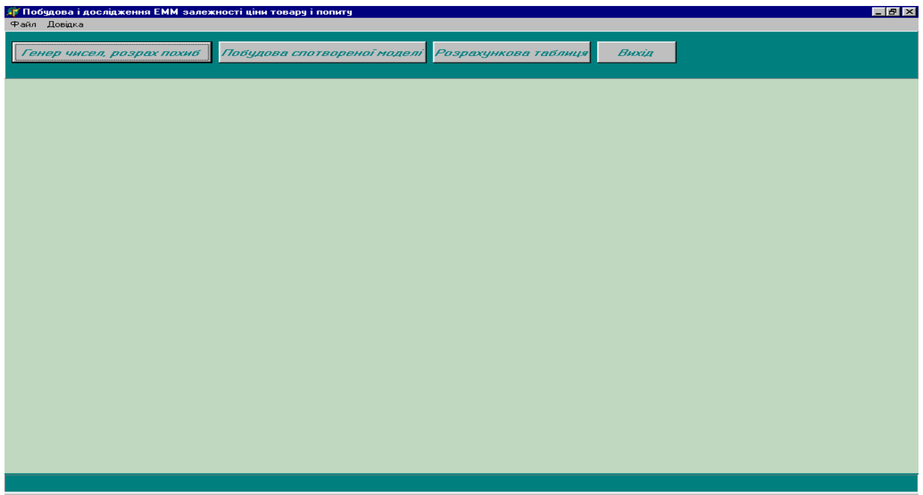
6.2. Опис програмного продукту

При розробці програмного продукту „Побудова і дослідження ЕММ залежності ціни товару і попиту на нього” створена головна управлінська форма, яка забезпечує виклик окремих програмних модулів:

1. генерування ;
2. побудова істинної моделі;
3. розрахункова таблиця;
4. довідка;
5. про програму.

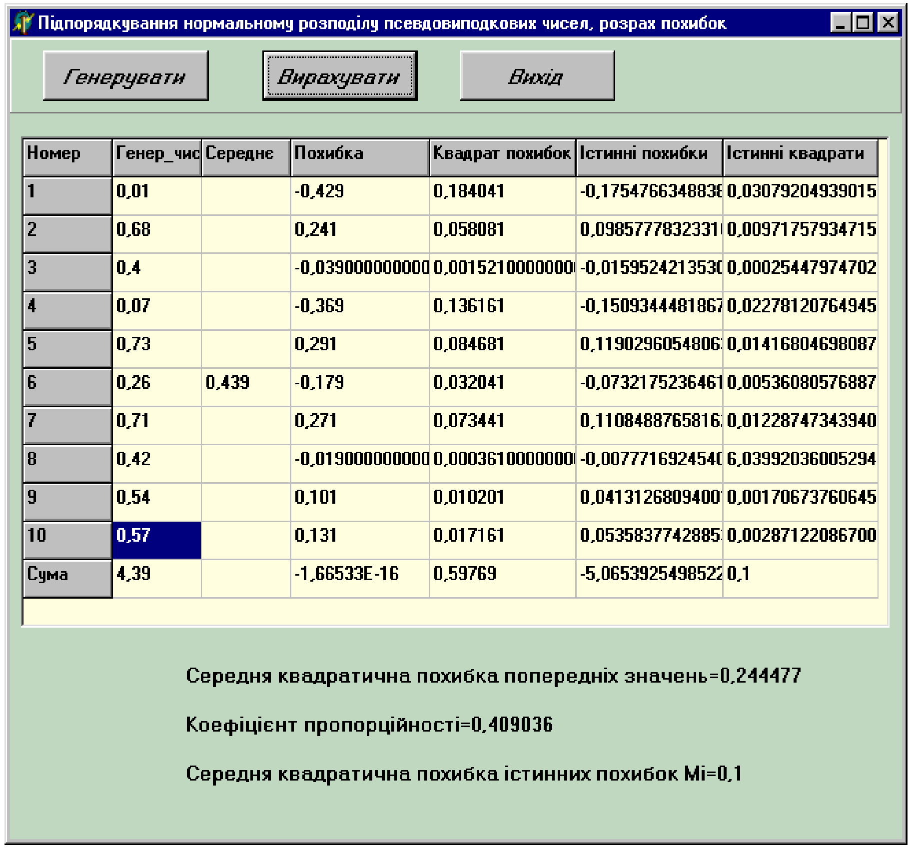
Модулі чітко визначені і кожний з них можна використовувати окремо. Кожний модуль має один вхід і один вихід , має не великий обсяг і може викликати тільки модулі беспосередньо наступного рівня. Модулі мають однаковий інтерфейс, що спрощує навчання роботи з програмою, і в подальшому вносить простоту і зручність в її використанні.

Головна форма має вигляд:



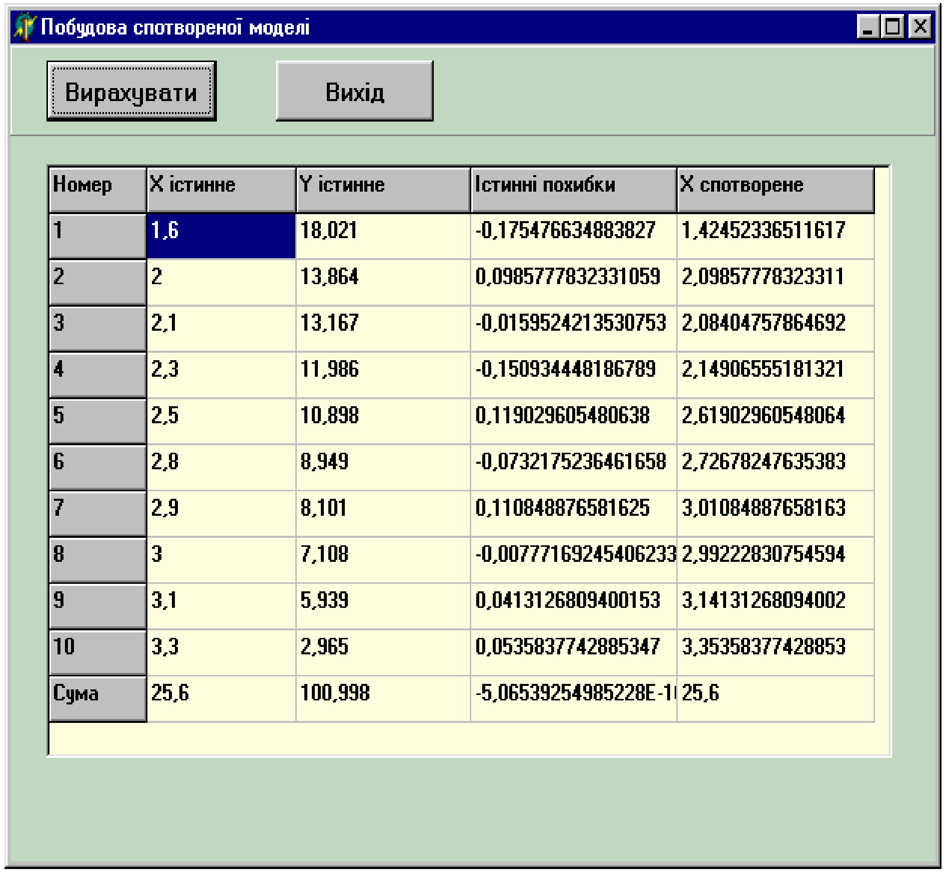
Функції програмних модулів:

1. Генерування чисел, розрахунок похибок:
   - генерування випадкових чисел;
   - підпорядкування випадкових чисел нормальному розподілу;
   - розрахунок попередніх значень істинних похибок, квадратів істинних похибок, середньоквадратичної похибки попередніх значень похибок, середньої квадратичної похибки істинних похибок.



Номер	Генер_чис	Середнє	Похибка	Квадрат похибок	Істинні похибки	Істинні квадрати
1	0,01		-0,429	0,184041	-0,175476634883	0,03079204939015
2	0,68		0,241	0,058081	0,098577783233	0,00971757934715
3	0,4		-0,039000000000	0,001521000000	-0,015952421353	0,00025447974702
4	0,07		-0,369	0,136161	-0,150934448186	0,02278120764945
5	0,73		0,291	0,084681	0,1190296054806	0,01416804698087
6	0,26	0,439	-0,179	0,032041	-0,0732175236461	0,00536080576887
7	0,71		0,271	0,073441	0,1108488765816	0,01228747343940
8	0,42		-0,019000000000	0,000361000000	-0,007771692454	6,03992036005294
9	0,54		0,101	0,010201	0,0413126809400	0,00170673760645
10	0,57		0,131	0,017161	0,0535837742885	0,00287122086700
Сума	4,39		-1,66533E-16	0,59769	-5,065392549852	0,1

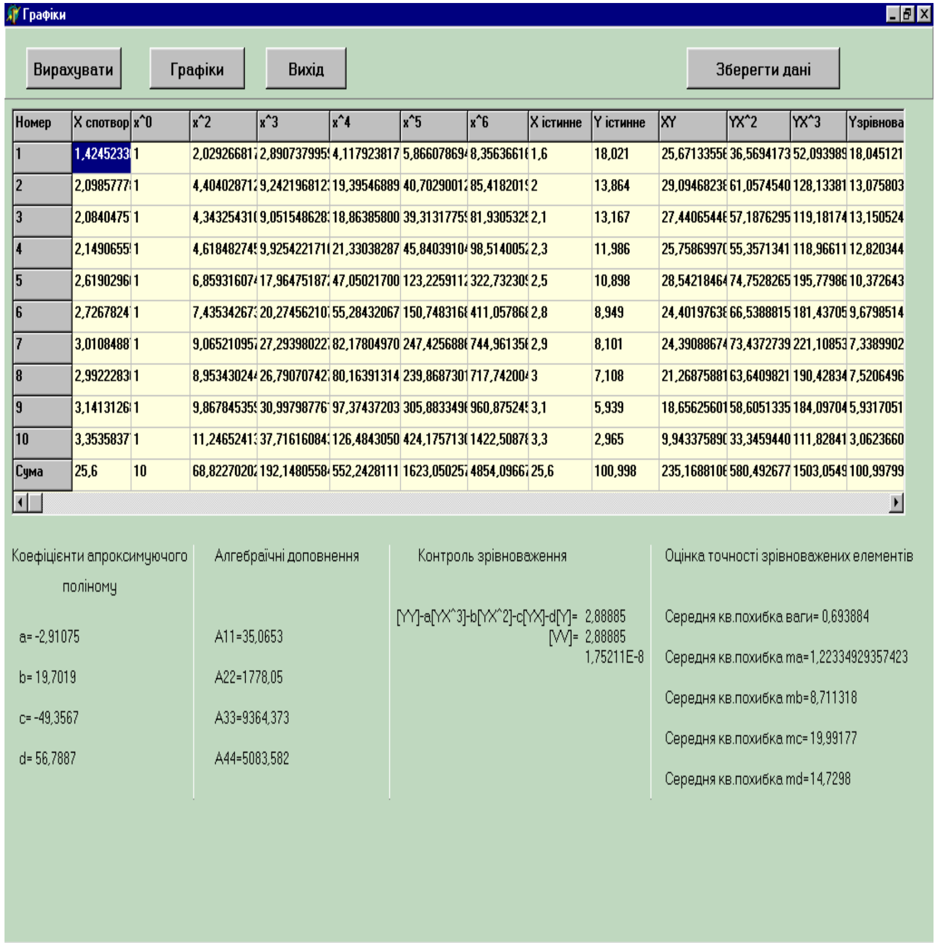
5. Побудова спотвореної моделі:
   - розраховуються X спотворене;



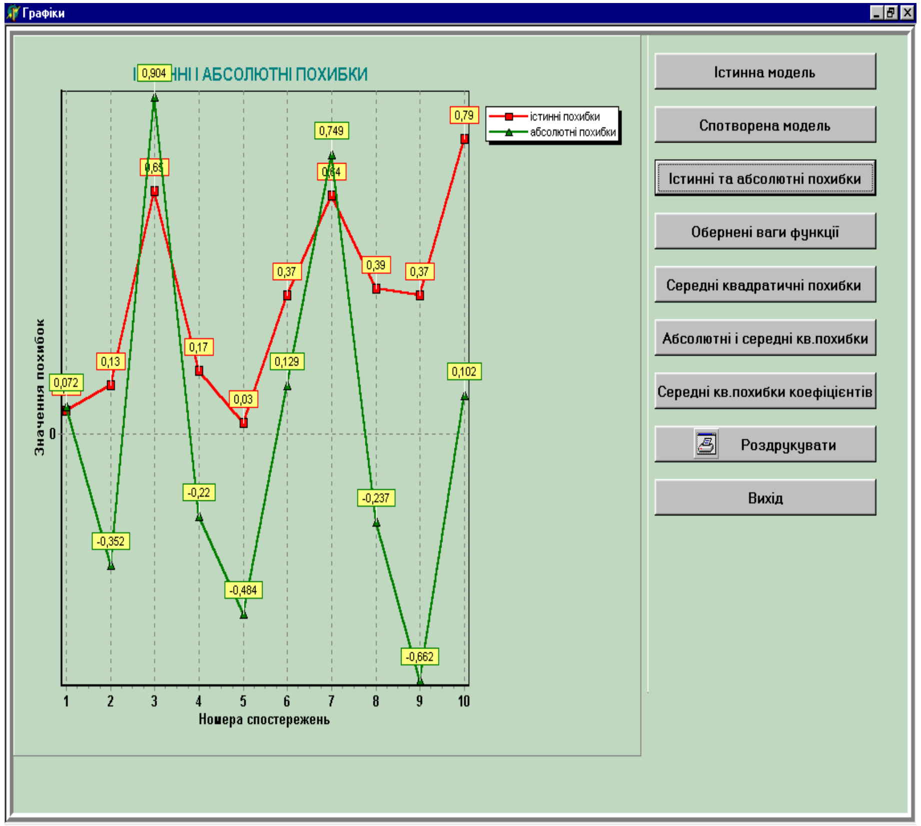
Номер	X істинне	Y істинне	Істинні похибки	X спотворене
1	1,6	18,021	-0,175476634883827	1,42452336511617
2	2	13,864	0,0985777832331059	2,09857778323311
3	2,1	13,167	-0,0159524213530753	2,08404757864692
4	2,3	11,986	-0,150934448186789	2,14906555181321
5	2,5	10,898	0,119029605480638	2,61902960548064
6	2,8	8,949	-0,0732175236461658	2,72678247635383
7	2,9	8,101	0,110848876581625	3,01084887658163
8	3	7,108	-0,00777169245406233	2,99222830754594
9	3,1	5,939	0,0413126809400153	3,14131268094002
10	3,3	2,965	0,0535837742885347	3,35358377428853
Сума	25,6	100,998	-5,06539254985228E-1	25,6

2. Розрахункова таблиця:
   - розрахунок x^0, x^2, x^3, x^4, x^5, x^6, xy, x^2y, x^3y, y зрівноважене;
   - розраховуються коефіцієнти апроксимуючого поліному;
   - робиться контроль зрівноваження;

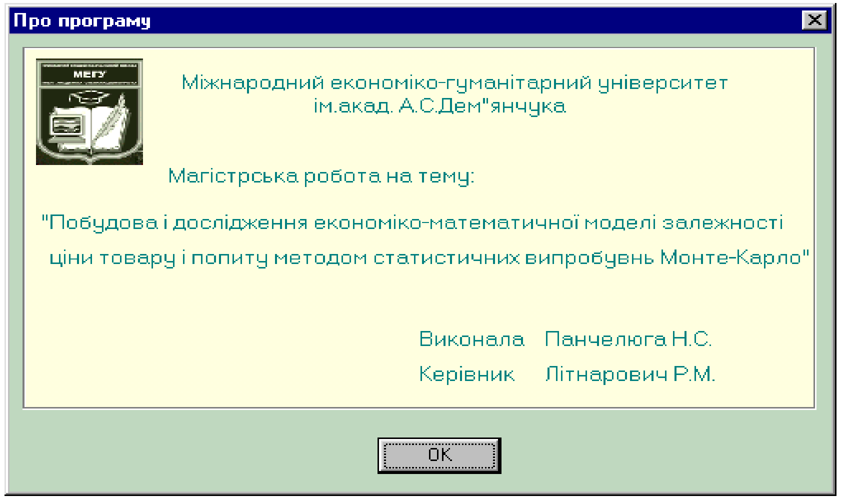
- проводиться оцінка точності зрівноважених елементів, а саме, розраховуються: середня квадратична похибка ваги, середні квадратичні похибки коефіцієнтів *a, b, c, d;*
- будуються графіки функцій;
- збереження даних.



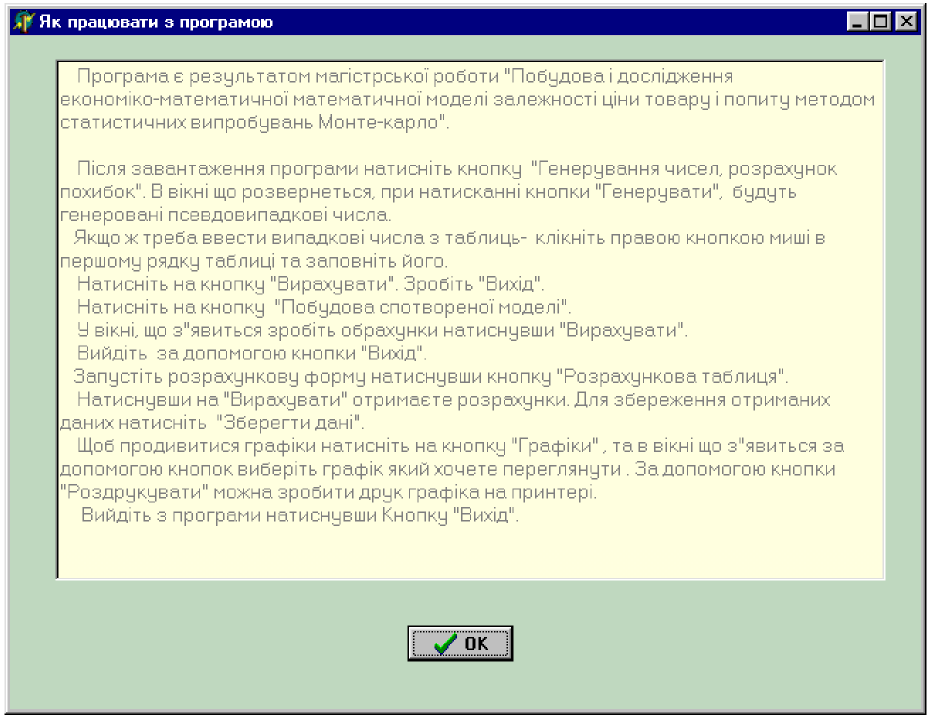
4. Будуються графіки:
- істинна модель;
- спотворена модель;
- істинні та абсолютні похибки;
- обернені ваги вункції:
- середні квадратичні похибки;
- абсолютні і середні квадратичні похибки;
- середні квадратичні похибки коефіцієнтів;
- друк графіків.



2. Про програму



3. Довідка



ВИСНОВКИ

На основі досліджень в даній роботі:

1. Генеровані випадкові числа, які приведено до нормованої досліджуваної точності.
2. На основі істинної моделі і генерованих істинних похибок побудована спотворена модель залежності впливу ціни товару на його попит.
3. Математична модель апроксимована за способом найменших квадратів кубічним поліномом.
4. Отримана формула

$$y = -2{,}911\,x^3 + 19{,}702\,x^2 - 49{,}357\,x + 56{,}789$$

залежності впливу ціни товару X на його попит Y.

5. Встановлено, що
   - середня квадратична похибка одиниці ваги за результатами зрівноваження складає 0,694 тисяч гривень;
   - середня квадратична похибка визначення коефіцієнта a при x^3 $m_a = 1{,}223$;
   - середня квадратична похибка визначення коефіцієнта b при x^2 $m_b = 8{,}711$;
   - середня квадратична похибка визначення коефіцієнта c при x $m_c = 19{,}992$;
   - середня квадратична похибка визначення коефіцієнта d $m_d = 14{,}729$;

середні квадратичні похибки зрівноваженої функції

mφ
0,692256875
0,394305165
0,398902959
0,378111569
0,345573
0,360047557
0,335053986
0,338796617
0,335111474
0,633020763

6. Розроблена методика підготовки істинних похибок наперед заданої точності.
7. Дана робота відкриває дорогу для проведення досліджень методом статистичних випробувань Монте Карло.
8. Вона дає можливість набрати велику статистику, тому що генеруються похибки індивідуально і вони не повторюються в інших моделях.

Результатом даного дипломного проекту є розробка програмного продукта. Розроблена програма дає можливість виконати необхідні розрахунки, що виникають при побудові економіко-математичної моделі, її апроксимації кубічним поліномом. Програма відповідає вимогам простоти, зручністі і дружності стосовно користувача, простота в освоєнні і не потребує спеціального навчання.

СПИСОК ТЕРМІНІВ ТА СКОРОЧЕНЬ

ГВЧ

Генератор випадкових чисел.

ГПВП

Генератор псевдо випадкової послідовності.

ГПВЧ

Генератор псевдовипадкових чисел.

ДВЧ

Датчик випадкових чисел.

ЕОМ

Електронно-обчислювальні машини.

ЕММ

Економіко-математичне моделювання .

МНК

Метод найменших квадратів.

РВП

Рівномірна випадкова послідовність.

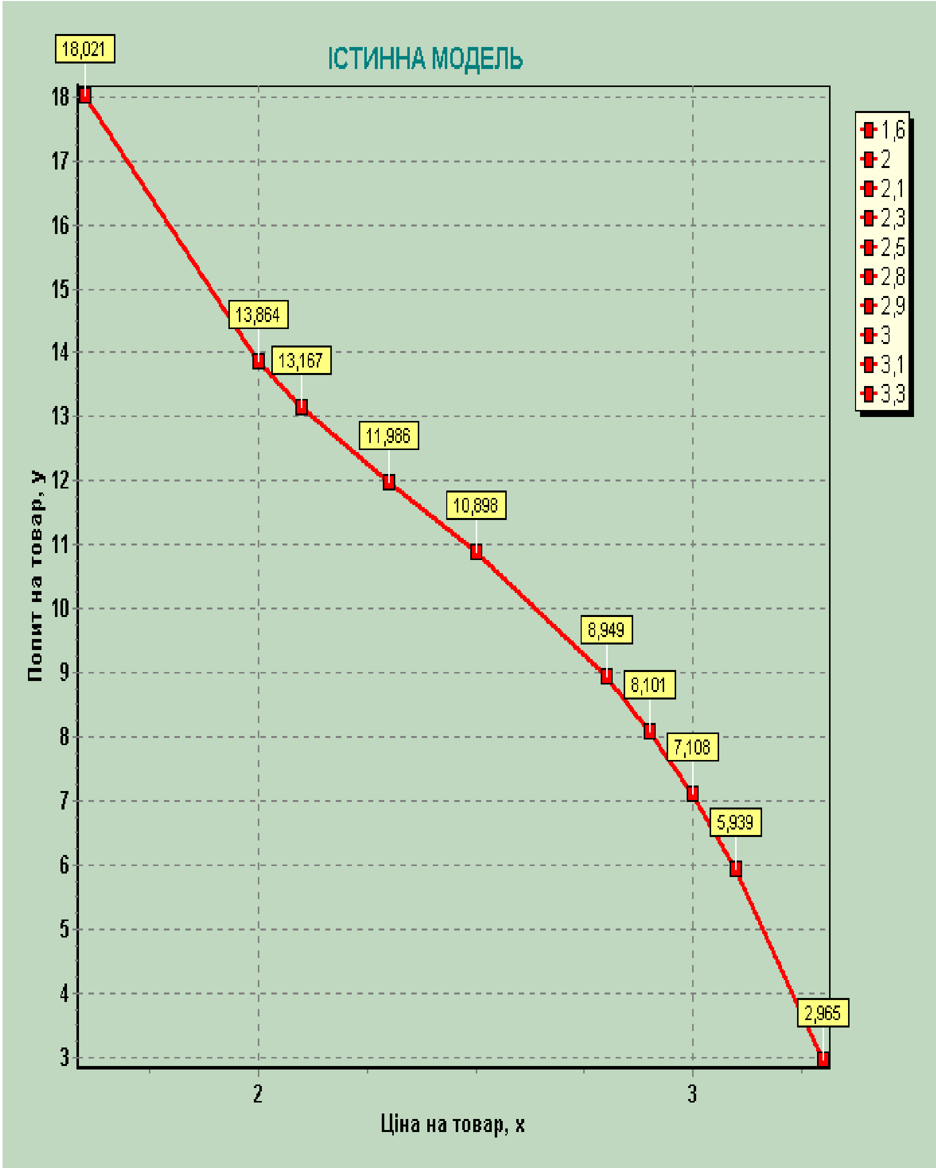
ЛІТЕРАТУРА ТА ДЖЕРЕЛА

1. В.В.Вітлінський. Моделювання економіки: Навч.посібник.-КНЕУ, 2003.-408с.
2. В.Ф.Ситник. Н.С.Орленко. Імітаційне моделювання: Навч.-метод. посібник для самост.вивч.дисц.-К.:КНЕУ, 1999.-208с.
3. П.Е.Данко. А.Г.Попов. Высшая математика в упражнениях и задачах. Ч.2. Изд.2-е. Учеб.пособие для втузов. М., Высшая школа», 1974.-464с.
4. В.А.Кудрявцев. Б.П.Демидович. Кратний курс высшей математики: Учебное пособие для вузов.-7-е изд., испр.-М.: Наука. Гл.ред.физ.-мат.лит., 1989.-656с.
5. Р.М.Літнарович. Основи наукових досліджень. Частина 1. Курс лекцій. МЕГУ, 2008.-75с.
6. Р.М.Літнарович. Алгебра матриць. Курс лекцій. МЕГУ,Рівне, 2007.-109 с.
7. Р.М.Літнарович. Конструювання і дослідження математичних моделей. Множинний аналіз.Частина 1. МЕГУ, Рівне, 2009.-127 с.
8. Р.М.Літнарович. Конструювання і дослідження математичних моделей. Поліноміальна апроксимація. Частина 2. МЕГУ, Рівне, 2009.-36 с.
9. Р.М.Літнарович Р.М. Конструювання і дослідження математичних моделей. Онтодидактика поліноміальної апроксимації. Частина 3. МЕГУ, Рівне, 2009.-32 с.
10. С.В. Глушаков, А.Л. Клєвцов. Програмування в середовищі Delphi 7.0. –Харків: Фоліо, 2003.- 528с.
11. В.Є. Гофман, А.Д. Хомоненко. Delphi 6.-2001.-1152с.

ДЖЕРЕЛА МЕРЕЖІ ІНТЕРНЕТ

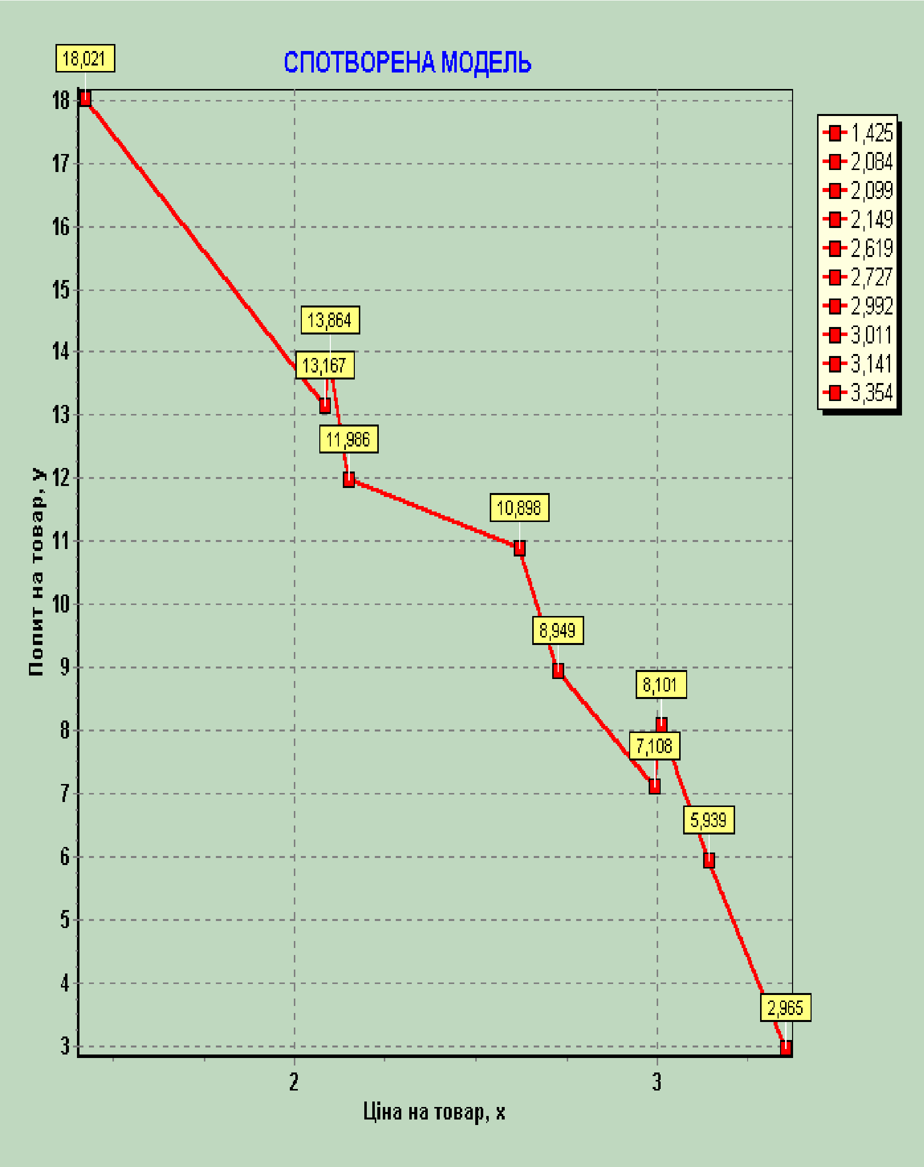
12. http://www.piter-press.ru
13. http://www.riskglossary.com.monte-karlo
14. http://www.devoid.com.ua/
15. http://www.programmersclub.ru

Додаток 1.



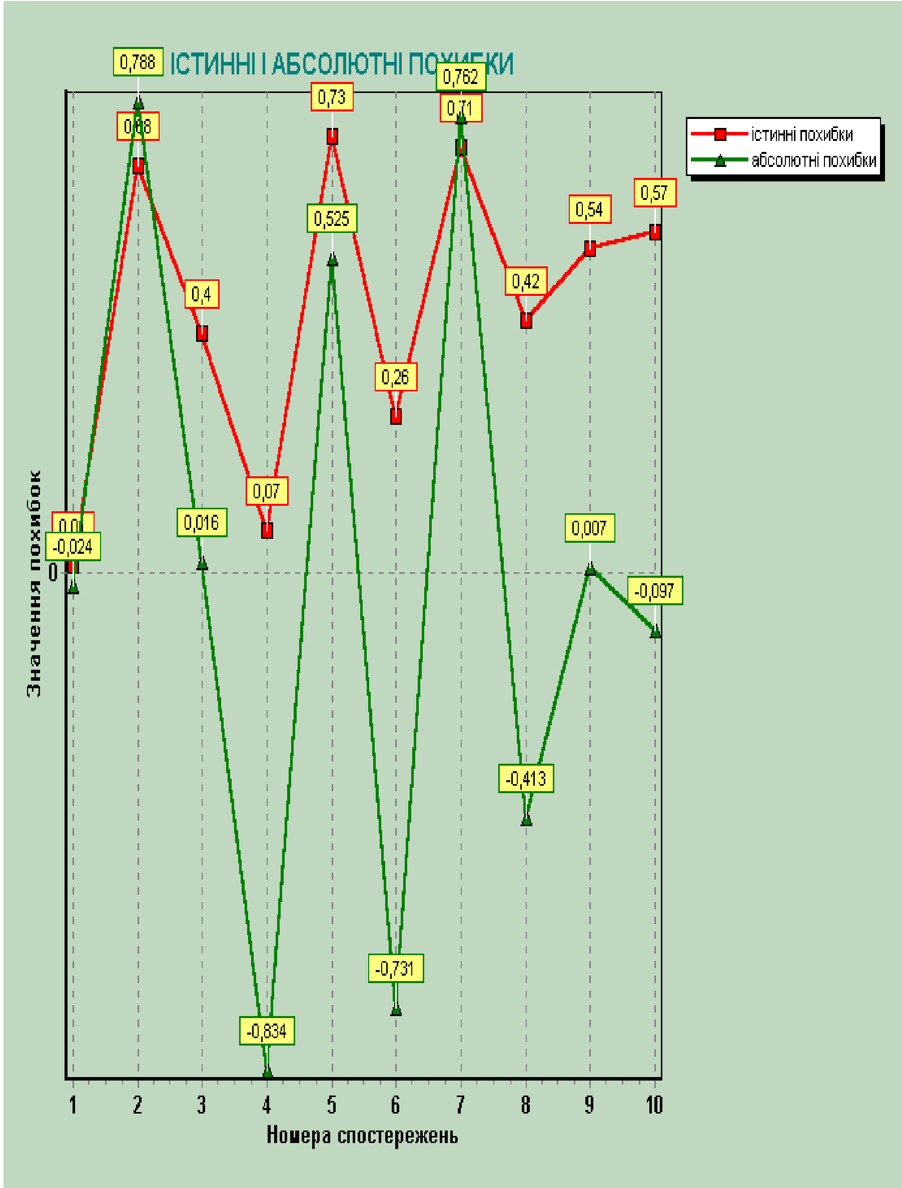
Мал.1. Графік істинної моделі

Додаток 2.

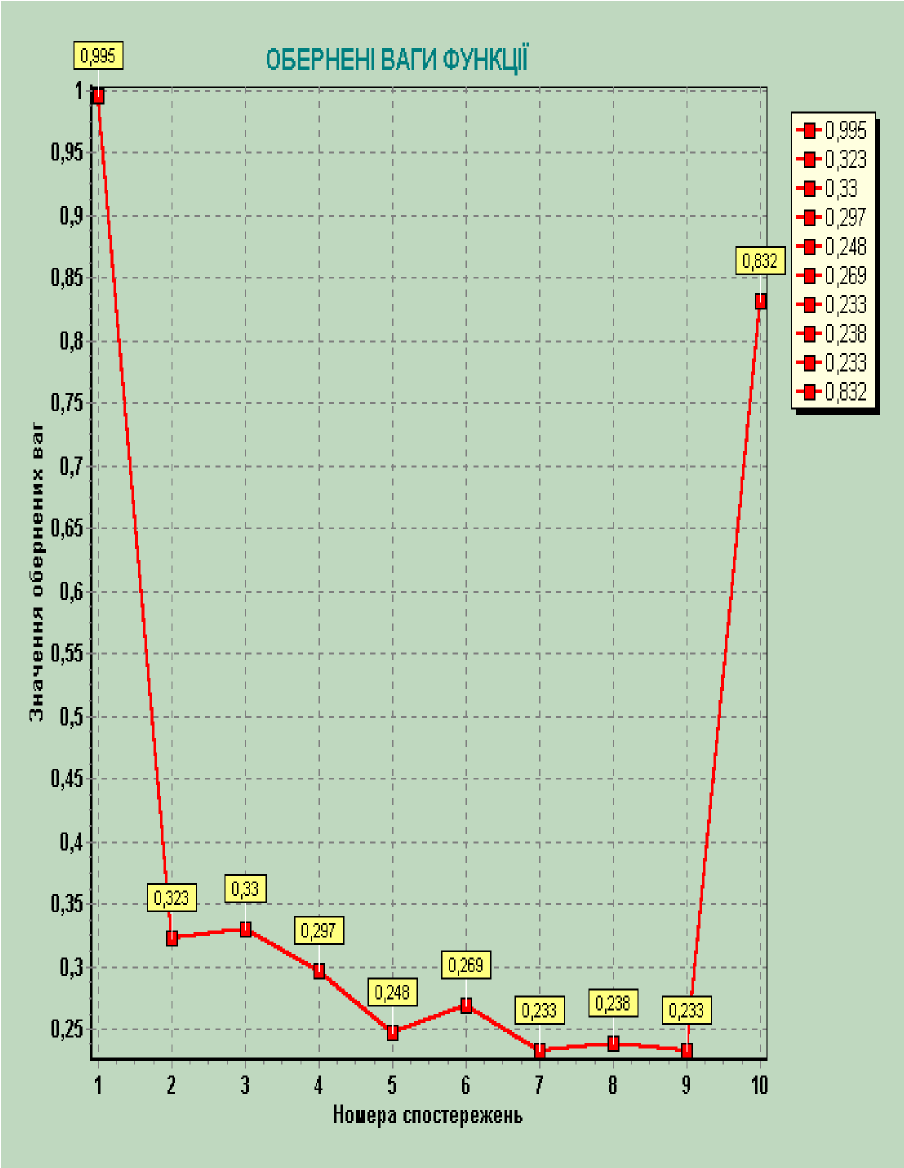


Мал.2. Графік спотвореної моделі

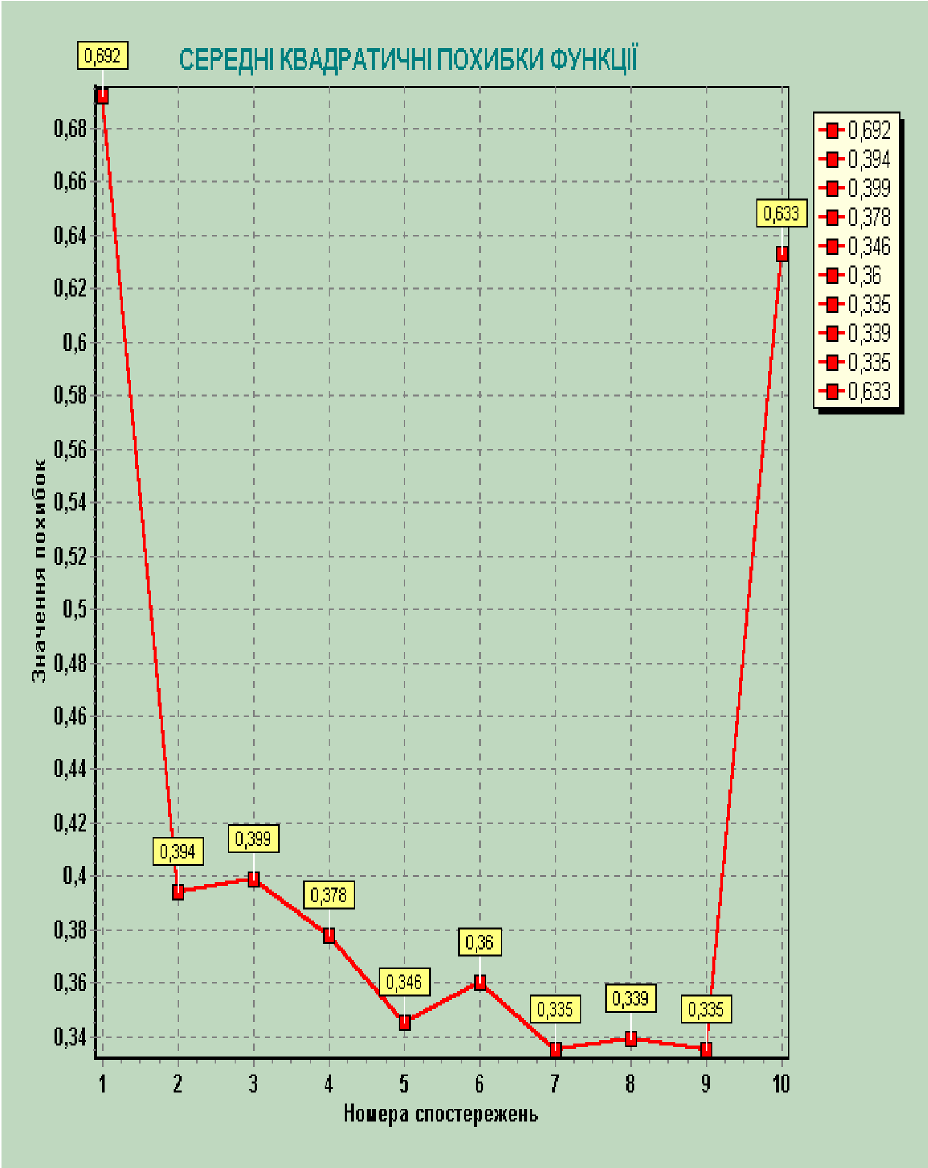
Додаток 3.



Мал.3. Графік істинних і абсолютних похибок

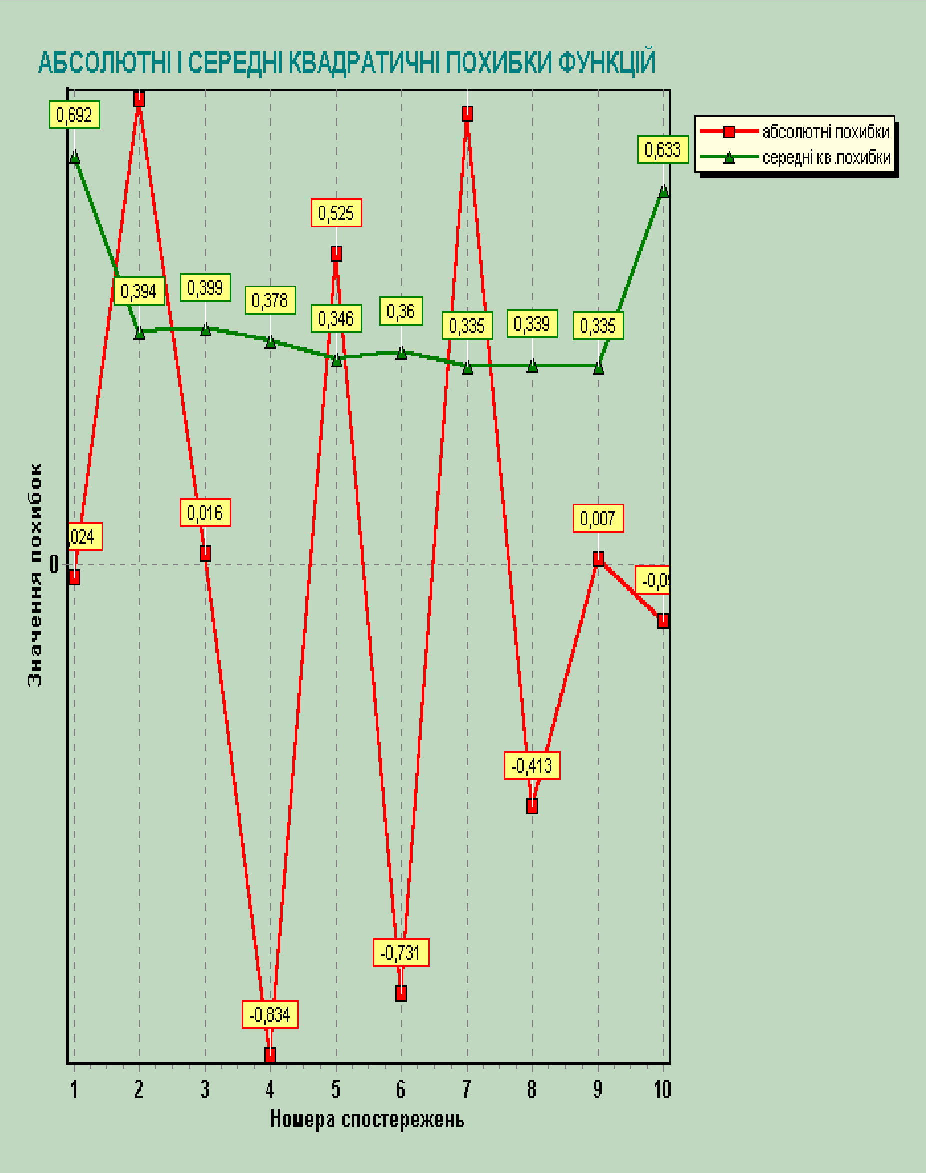


Мал.4. Графік обернених ваг



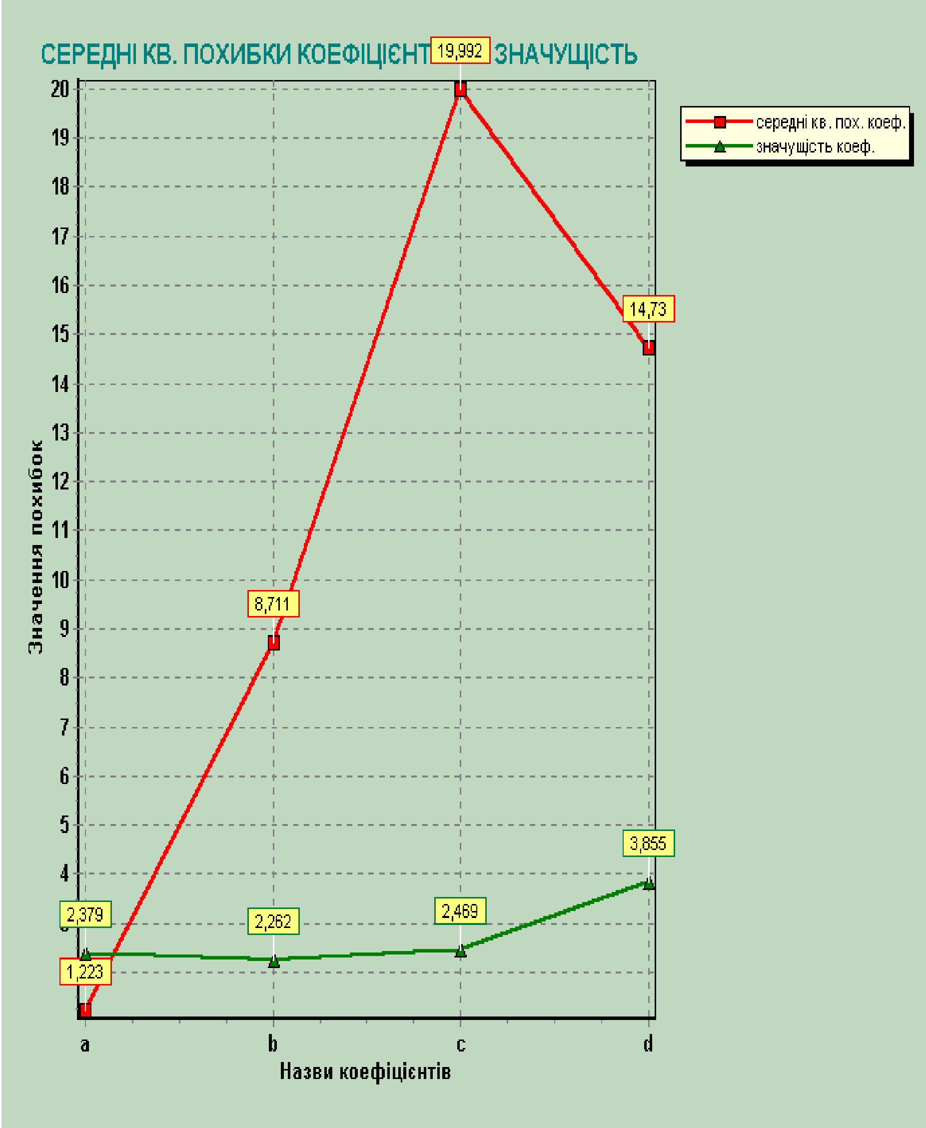
Мал.5. Графік середніх квадратичних похибок функції

Додаток 6.



Мал.6. Графік абсолютних і середніх квадратичних похибок функції

Додаток 7.



Мал.7. Графік середніх квадратичних похибок коефіцієнтів та їх значущості

Додаток 8.

```
unit UMain;

.........................


procedure TForm1.N2Click(Sender: TObject);
begin
  Close;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
Application.CreateForm(TFgener,Fgener);
Fgener.ShowModal;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
Form1.Close;
end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
 Application.CreateForm(TFModel,FModel);
 FModel.ShowModal;
end;

procedure TForm1.BitBtn4Click(Sender: TObject);
begin
  Application.CreateForm(TFRozr,FRozr);
  FRozr.ShowModal;
end;

procedure TForm1.N5Click(Sender: TObject);
begin
   Application.CreateForm(TAboutBox,AboutBox);
 AboutBox.ShowModal;
    end
```


Додаток 9.

```
unit Ugener;
..........................

procedure TFgener.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
 action:=CaFree;
end;

procedure TFgener.FormActivate(Sender: TObject);
begin
tabl.Cells[0,0]:='Номер';
 tabl.Cells[1,0]:='Генер_число';
 tabl.Cells[2,0]:='Середнє';
 tabl.Cells[3,0]:='Похибка';
 tabl.Cells[4,0]:='Квадрат похибок';
 tabl.Cells[5,0]:='Істинні похибки';
 tabl.Cells[6,0]:='Істинні квадрати';
 tabl.Cells[0,1]:='1';
 tabl.Cells[0,2]:='2';
 tabl.Cells[0,3]:='3';
 tabl.Cells[0,4]:='4';
 tabl.Cells[0,5]:='5';
 tabl.Cells[0,6]:='6';
 tabl.Cells[0,7]:='7';
 tabl.Cells[0,8]:='8';
 tabl.Cells[0,9]:='9';
 tabl.Cells[0,10]:='10';
 tabl.Cells[0,11]:='Сума';
end;

procedure TFgener.BitBtn1Click(Sender: TObject);
var
a:array [1..10] of real;
summ:real;
sr,b,d,v:real;
i,j:integer;
   f,w:TextFile;
   istPohib:string[23];
```

```
generPohib:string[23];
begin

for i:=1 to 10 do
 if tabl.Cells[1,i]<>''
  then
   a[i]:=StrToFloat(tabl.Cells[1,i])
    else
        Exit;
     for i:=1 to 10 do
       summ:=summ+a[i];
       sr:=summ/10;
  tabl.Cells[1,11]:=FloatToStr(summ);
  tabl.Cells[2,6]:=FloatToStr(sr);
             begin
              generPohib:=tabl.Cells[1,i];
              AssignFile(w,'date\generPohib.txt');
              Rewrite(w);
               for i:=0 to 12 do
              writeln(w,tabl.Cells[1,i]);
              CloseFile(w);
             end;
  tabl.Cells[3,1]:=floattostr((strtofloat(tabl.Cells[1,1]))-sr);
  tabl.Cells[3,2]:=floattostr((strtofloat(tabl.Cells[1,2]))-sr);;
  tabl.Cells[3,3]:=floattostr((strtofloat(tabl.Cells[1,3]))-sr);
  tabl.Cells[3,4]:=floattostr((strtofloat(tabl.Cells[1,4]))-sr);
  tabl.Cells[3,5]:=floattostr((strtofloat(tabl.Cells[1,5]))-sr);
  tabl.Cells[3,6]:=floattostr((strtofloat(tabl.Cells[1,6]))-sr
tabl.Cells[3,7]:=floattostr((strtofloat(tabl.Cells[1,7]))-sr
tabl.Cells[3,8]:=floattostr((strtofloat(tabl.Cells[1,8]))-sr
tabl.Cells[3,9]:=floattostr((strtofloat(tabl.Cells[1,9]))-sr
tabl.Cells[3,10]:=floattostr((strtofloat(tabl.Cells[1,10]))-sr
       for i:=1 to 10 do
 if tabl.Cells[3,i]<>''
  then
   a[i]:=StrToFloat(tabl.Cells[3,i])
    else a[i]:=0;
     summ:=0;
     for i:=1 to 10 do
       summ:=summ+a[i];
```



```
tabl.Cells[3,11]:=FloatToStrf(summ,ffGeneral,6,5);
  tabl.Cells[4,1]:=floattostr(sqr(strtofloat(tabl.Cells[3,1])));
  tabl.Cells[4,2]:=floattostr(sqr(strtofloat(tabl.Cells[3,2])));
  tabl.Cells[4,3]:=floattostr(sqr(strtofloat(tabl.Cells[3,3])));
  tabl.Cells[4,4]:=floattostr(sqr(strtofloat(tabl.Cells[3,4])));
  tabl.Cells[4,5]:=floattostr(sqr(strtofloat(tabl.Cells[3,5])));
  tabl.Cells[4,6]:=floattostr(sqr(strtofloat(tabl.Cells[3,6])));
  tabl.Cells[4,7]:=floattostr(sqr(strtofloat(tabl.Cells[3,7])));
  tabl.Cells[4,8]:=floattostr(sqr(strtofloat(tabl.Cells[3,8])));
  tabl.Cells[4,9]:=floattostr(sqr(strtofloat(tabl.Cells[3,9])));;
  tabl.Cells[4,10]:=floattostr(sqr(strtofloat(tabl.Cells[3,10])));
       for i:=1 to 10 do
  if tabl.Cells[4,i]<>''
   then
   a[i]:=StrToFloat(tabl.Cells[4,i])
    else a[i]:=0;
     summ:=0;
     for i:=1 to 10 do
       summ:=summ+a[i];
   tabl.Cells[4,11]:=FloatToStr(summ
         b:=sqrt((strtofloat(tabl.Cells[4,11]))/10);
         d:=0.1/b;
         label1.Caption:='Середня квадратична похибка попередніх
значень='+FloatToStrf(b,ffGeneral,6,5);
         label2.Caption:='Коефіцієнт
пропорційності='+FloatToStrf(d,ffGeneral,6,5);

  tabl.Cells[5,1]:=floattostr(d*(strtofloat(tabl.Cells[3,1])));
  tabl.Cells[5,2]:=floattostr(d*(strtofloat(tabl.Cells[3,2])));
  tabl.Cells[5,3]:=floattostr(d*(strtofloat(tabl.Cells[3,3])));
  tabl.Cells[5,4]:=floattostr(d*(strtofloat(tabl.Cells[3,4])))
  tabl.Cells[5,5]:=floattostr(d*(strtofloat(tabl.Cells[3,5])));
tabl.Cells[5,6]:=floattostr(d*(strtofloat(tabl.Cells[3,6])));/
  tabl.Cells[5,7]:=floattostr(d*(strtofloat(tabl.Cells[3,7])));/
  tabl.Cells[5,8]:=floattostr(d*(strtofloat(tabl.Cells[3,8])));
  tabl.Cells[5,9]:=floattostr(d*(strtofloat(tabl.Cells[3,9])));/
  tabl.Cells[5,10]:=floattostr(d*(strtofloat(tabl.Cells[3,10])));
      for i:=1 to 10 do
   if tabl.Cells[5,i]<>''
    then
```

```
a[i]:=StrToFloat(tabl.Cells[5,i])
     else a[i]:=0;
     summ:=0;
     for i:=1 to 10 do
       summ:=summ+a[i];
   tabl.Cells[5,11]:=FloatToStr(summ);//,ffGeneral,6,5);
tabl.Cells[6,1]:=floattostr(sqr(strtofloat(tabl.Cells[5,1])))
  tabl.Cells[6,2]:=floattostr(sqr(strtofloat(tabl.Cells[5,2])));
  tabl.Cells[6,3]:=floattostr(sqr(strtofloat(tabl.Cells[5,3])));/
  tabl.Cells[6,4]:=floattostr(sqr(strtofloat(tabl.Cells[5,4])));/
  tabl.Cells[6,5]:=floattostr(sqr(strtofloat(tabl.Cells[5,5])));/
  tabl.Cells[6,6]:=floattostr(sqr(strtofloat(tabl.Cells[5,6])));
  tabl.Cells[6,7]:=floattostr(sqr(strtofloat(tabl.Cells[5,7])));/
tabl.Cells[6,8]:=floattostr(sqr(strtofloat(tabl.Cells[5,8])));
  tabl.Cells[6,9]:=floattostr(sqr(strtofloat(tabl.Cells[5,9])));
tabl.Cells[6,10]:=floattostr(sqr(strtofloat(tabl.Cells[5,10])));
       for i:=1 to 10 do
   if tabl.Cells[6,i]<>''
    then
    a[i]:=StrToFloat(tabl.Cells[6,i])
      else a[i]:=0;
     summ:=0;
     for i:=1 to 10 do
       summ:=summ+a[i];
   tabl.Cells[6,11]:=FloatToStrf(summ,ffGeneral,6,5);
   label3.Caption:='Середня квадратична похибка істинних похибок
Mi='+(tabl.cells[6,11]);
               begin
               istPohib:=tabl.Cells[5,i];
               AssignFile(f,'date\istPohib.txt');
               Rewrite(f);
                for i:=0 to 12 do
               writeln(f,tabl.Cells[5,i]);
               CloseFile(f);
               end;
  end;

procedure TFgener.BitBtn2Click(Sender: TObject);
begin
Fgener.Close;
```



```
end;


procedure TFgener.tablKeyPress(Sender: TObject; var Key: Char);
var
i:integer;
begin
case key of
#8,'0'..'9':;
#13:;
{if tabl.Col<tabl.ColCount-1
  then tabl.Col:=tabl.Col+1;  }

  '.',',':
   begin
   if key<>DecimalSeparator then
    key:=DecimalSeparator;
        end;
        '-':
   if (tabl.Cells[1,i])<>''
    then key:=Chr(0);
     else
      key:=Chr(0);
end;
end;

procedure TFgener.BitBtn3Click(Sender: TObject);
var
i,j:integer;
a:array [1..10] of real;
begin
randomize();
for i:=1 to 10 do
a[i]:=strtofloat('0,'+IntToStr(random(10))+IntToStr(random(10)));
    for i:=1 to 10{stringgrid1.rowCount} do   tabl.cells[1,i]:=floattostr(a[i]);
end;  end.
```

Додаток 10.

```
unit UModel;
..............................

procedure TFModel.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
 action:=CaFree;
end;

procedure TFModel.FormActivate(Sender: TObject);
begin
 tabl.Cells[0,0]:='Номер';
 tabl.Cells[1,0]:='X істинне';
 tabl.Cells[2,0]:='Y істинне';
 tabl.Cells[3,0]:='Істинні похибки';
 tabl.Cells[4,0]:='X спотворене';
 tabl.Cells[0,1]:='1';
 tabl.Cells[0,2]:='2';
 tabl.Cells[0,3]:='3';
 tabl.Cells[0,4]:='4';
 tabl.Cells[0,5]:='5';
 tabl.Cells[0,6]:='6';
 tabl.Cells[0,7]:='7';
 tabl.Cells[0,8]:='8';
 tabl.Cells[0,9]:='9';
 tabl.Cells[0,10]:='10';
 tabl.Cells[0,11]:='Сума';
end;

procedure TFModel.BitBtn1Click(Sender: TObject);
var
 a:array [1..10] of real;
 summ:real;
 sr:real;
 i:integer;
 f:TextFile;
 xspotv:string[23];
begin
 tabl.Cols[1].LoadFromFile('date\Xist.txt');
```



```
tabl.Cols[2].LoadFromFile('date\Yist.txt');
 tabl.Cols[3].LoadFromFile('date\istPohib.txt');
tabl.Cells[4,1]:=floattostr((strtofloat(tabl.Cells[1,1]))+(strtofloat(tabl.
Cells[3,1])
tabl.Cells[4,2]:=floattostr((strtofloat(tabl.Cells[1,2]))+(strtofloat(tabl.
Cells[3,2])
tabl.Cells[4,3]:=floattostr((strtofloat(tabl.Cells[1,3]))+(strtofloat(tabl.
Cells[3,3tabl.Cells[4,4]:=floattostr((strtofloat(tabl.Cells[1,4]))+(strtof
loat(tabl.Cells[3,4tabl.Cells[4,5]:=floattostr((strtofloat(tabl.Cells[1,5]
))+(strtofloat(tabl.Cells[3,5tabl.Cells[4,6]:=floattostr((strtofloat(tabl.
Cells[1,6]))+(strtofloat(tabl.Cells[3,6tabl.Cells[4,7]:=floattostr((strtof
loat(tabl.Cells[1,7]))+(strtofloat(tabl.Cells[3,7])));//,
tabl.Cells[4,8]:=floattostr((strtofloat(tabl.Cells[1,8]))+(strtofloat(tabl.
Cells[3,8tabl.Cells[4,9]:=floattostr((strtofloat(tabl.Cells[1,9]))+(strtof
loat(tabl.Cells[3,9tabl.Cells[4,10]:=floattostr((strtofloat(tabl.Cells[1,1
0]))+(strtofloat(tabl.Cells[3,10])));//,ffGeneral,6,5);
     for i:=1 to 10 do
      if tabl.Cells[4,i]<>''
       then
      a[i]:=StrToFloat(tabl.Cells[4,i])
       else a[i]:=0;
       summ:=0;
        for i:=1 to 10 do
        summ:=summ+a[i];
    tabl.Cells[4,11]:=FloatToStr(summ);
    for i:=1 to 10 do
     if tabl.Cells[1,i]<>''
      then
     a[i]:=StrToFloat(tabl.Cells[1,i])
      else a[i]:=0;
      summ:=0;
      for i:=1 to 10 do
      summ:=summ+a[i];
    tabl.Cells[1,11]:=FloatToStr(summ);
     for i:=1 to 10 do
      if tabl.Cells[2,i]<>''
```

```
    then
   a[i]:=StrToFloat(tabl.Cells[2,i])
     else a[i]:=0;
     summ:=0;
     for i:=1 to 10 do
     summ:=summ+a[i];



    tabl.Cells[2,11]:=FloatToStr(summ);
       begin
           xspotv:=tabl.Cells[4,i];
           AssignFile(f,'date\Xspotv.txt');
           Rewrite(f);
          for i:=0 to 12 do
           writeln(f,tabl.Cells[4,i]);
           CloseFile(f);
       end;
end;

procedure TFModel.BitBtn2Click(Sender: TObject);
begin
  FModel.Close;
end;

end.
```


Додаток 11.

```
unit URozr;

.........................


procedure TFRozr.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=CaFree;
end;

procedure TFRozr.FormActivate(Sender: TObject);
begin
 tabl.Cells[0,0]:='Номер';
 tabl.Cells[1,0]:='X спотворене';
 tabl.Cells[2,0]:='x^0';
 tabl.Cells[3,0]:='x^2';
 tabl.Cells[4,0]:='x^3';
 tabl.Cells[5,0]:='x^4';
 tabl.Cells[6,0]:='x^5';
 tabl.Cells[7,0]:='x^6';
 tabl.Cells[8,0]:='Xістинне';
 tabl.Cells[9,0]:='Yістинне';
 tabl.Cells[10,0]:='XY';
 tabl.Cells[11,0]:='YX^2';
 tabl.Cells[12,0]:='YX^3';
 tabl.Cells[13,0]:='Yзрівноважене';
 tabl.Cells[14,0]:='Yi-Yзрівн.';
 tabl.Cells[15,0]:='VV';
 tabl.Cells[16,0]:='YY';
 tabl.Cells[17,0]:='Генер_число';
 tabl.Cells[18,0]:='Вектор_оберн_ваг';
 tabl.Cells[19,0]:='Корінь ваг';
 tabl.Cells[20,0]:='Сер_кв_пох функції';
 tabl.Cells[0,1]:='1';
 tabl.Cells[0,2]:='2';
 tabl.Cells[0,3]:='3';
 tabl.Cells[0,4]:='4';
 tabl.Cells[0,5]:='5';
```

```
tabl.Cells[0,6]:='6';
 tabl.Cells[0,7]:='7';
 tabl.Cells[0,8]:='8';
 tabl.Cells[0,9]:='9';
 tabl.Cells[0,10]:='10';
 tabl.Cells[0,11]:='Сума';
 BitBtn3.Enabled:=false;
end;

procedure TFRozr.BitBtn1Click(Sender: TObject);
 var
  a:array [1..10] of real;
  summ:real;
sr,n28,n21,n35,n42,n49,d,d1,d2,d3,o,n,m,l,k,g,j,r,q,p,h,c,a11,a22,a33,a44,h
54,h56,mv,ma,mb,mc,md,e,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x1
4,x15,x16,y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,s31,s32,s33,s34,s35,s36,s37,s38,
s39,s3,s21,s22,s23,s24,s25,s26,s27,s28,s29,s2,ss1,ss2,ss3,ss4,ss5,ss6,ss7,ss
8,ss9,ss,s01,s02,s03,s04,s05,s06,s07,s08,s09,s0,zn1,zn2,zn3,
zn4:real;
  i:integer;
begin
  Panel2.Visible:=false;
  tabl.Cols[1].LoadFromFile('date\Xspotv.txt');
  tabl.Cols[8].LoadFromFile('date\Xist.txt');
  tabl.Cols[9].LoadFromFile('date\Yist.txt');
  tabl.Cols[17].LoadFromFile('date\generPohib.txt');
  tabl.Cells[2,1]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,1])))));
  tabl.Cells[2,2]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,2])))));
  tabl.Cells[2,3]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,3])))));/
  tabl.Cells[2,4]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,4])))));/
tabl.Cells[2,5]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,5])))));/
tabl.Cells[2,6]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,6])))));//
tabl.Cells[2,7]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,7])))));/
tabl.Cells[2,8]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,8])))));/
  tabl.Cells[2,9]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,9])))));
tabl.Cells[2,10]:=floattostr(exp(0*(ln(strtofloat(tabl.Cells[1,10])))));
   for i:=1 to 10 do
        if tabl.Cells[2,i]<>''
         then
          a[i]:=StrToFloat(tabl.Cells[2,i])
```



```
else a[i]:=0;
           summ:=0;
              for i:=1 to 10 do
              summ:=summ+a[i];
   tabl.Cells[2,11]:=FloatToStr(summ);//,ffGeneral,6,5);
  tabl.Cells[3,1]:=floattostr(sqr(strtofloat(tabl.Cells[1,1])));/
tabl.Cells[3,2]:=floattostr(sqr(strtofloat(tabl.Cells[1,2])));//
tabl.Cells[3,3]:=floattostr(sqr(strtofloat(tabl.Cells[1,3])));//
tabl.Cells[3,4]:=floattostr(sqr(strtofloat(tabl.Cells[1,4])));
tabl.Cells[3,5]:=floattostr(sqr(strtofloat(tabl.Cells[1,5])));
tabl.Cells[3,6]:=floattostr(sqr(strtofloat(tabl.Cells[1,6])));
tabl.Cells[3,7]:=floattostr(sqr(strtofloat(tabl.Cells[1,7])));/
tabl.Cells[3,8]:=floattostr(sqr(strtofloat(tabl.Cells[1,8])));
tabl.Cells[3,9]:=floattostr(sqr(strtofloat(tabl.Cells[1,9])));
  tabl.Cells[3,10]:=floattostr(sqr(strtofloat(tabl.Cells[1,10])))
       for i:=1 to 10 do
        if tabl.Cells[3,i]<>''
         then
          a[i]:=StrToFloat(tabl.Cells[3,i])
           else a[i]:=0;
           summ:=0;
            for i:=1 to 10 do
            summ:=summ+a[i];
   tabl.Cells[3,11]:=FloatToStr(summ);//,ffGeneral,6,5);
tabl.Cells[4,1]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,1])))));
  tabl.Cells[4,2]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,2])))));
  tabl.Cells[4,3]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,3])))));
  tabl.Cells[4,4]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,4])))));/;
  tabl.Cells[4,5]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,5])))));/;
  tabl.Cells[4,6]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,6])))));;
  tabl.Cells[4,7]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,7])))));/
tabl.Cells[4,8]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,8])))));/
  tabl.Cells[4,9]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,9])))));/;
  tabl.Cells[4,10]:=floattostr(exp(3*(ln(strtofloat(tabl.Cells[1,10])))));;
       for i:=1 to 10 do
        if tabl.Cells[4,i]<>''
         then
          a[i]:=StrToFloat(tabl.Cells[4,i])
           else a[i]:=0;
            summ:=0;
```

```
for i:=1 to 10 do
            summ:=summ+a[i];
   tabl.Cells[4,11]:=FloatToStr(summ);//,ffGeneral,6,5);
tabl.Cells[5,1]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,1])))));;
  tabl.Cells[5,2]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,2])))));
  tabl.Cells[5,3]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,3])))));
  tabl.Cells[5,4]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,4]))))));
  tabl.Cells[5,5]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,5])))));/
tabl.Cells[5,6]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,6])))));
  tabl.Cells[5,7]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,7]))))));
  tabl.Cells[5,8]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,8]))))));
tabl.Cells[5,9]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,9]))))));/
tabl.Cells[5,10]:=floattostr(exp(4*(ln(strtofloat(tabl.Cells[1,10])))))
  for i:=1 to 10 do
        if tabl.Cells[5,i]<>''
         then
          a[i]:=StrToFloat(tabl.Cells[5,i])
           else a[i]:=0;
            summ:=0;
             for i:=1 to 10 do
             summ:=summ+a[i];
   tabl.Cells[5,11]:=FloatToStr(summ);//,ffGeneral,6,5);
tabl.Cells[6,1]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,1]))))));
  tabl.Cells[6,2]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,2])))));
tabl.Cells[6,3]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,3])))));
  tabl.Cells[6,4]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,4])))))
.Cells[6,5]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,5]))))));
  tabl.Cells[6,6]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,6])))));
  tabl.Cells[6,7]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,7])))));
tabl.Cells[6,8]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,8]))))));
  tabl.Cells[6,9]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,9])))));
  tabl.Cells[6,10]:=floattostr(exp(5*(ln(strtofloat(tabl.Cells[1,10])))));
       for i:=1 to 10 do
        if tabl.Cells[6,i]<>''
         then
          a[i]:=StrToFloat(tabl.Cells[6,i])
           else a[i]:=0;
            summ:=0;
             for i:=1 to 10 do
             summ:=summ+a[i];
```



```
tabl.Cells[6,11]:=FloatToStr(summ);//,ffGeneral,7,6);
tabl.Cells[7,1]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,1])))));/
tabl.Cells[7,2]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,2])))));;
  tabl.Cells[7,3]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,3])))));;
  tabl.Cells[7,4]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,4])))));
  tabl.Cells[7,5]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,5])))));
  tabl.Cells[7,6]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,6])))));
  tabl.Cells[7,7]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,7])))));
  tabl.Cells[7,8]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,8])))));/
tabl.Cells[7,9]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,9])))));
  tabl.Cells[7,10]:=floattostr(exp(6*(ln(strtofloat(tabl.Cells[1,10])))))
       for i:=1 to 10 do
        if tabl.Cells[7,i]<>''
         then
          a[i]:=StrToFloat(tabl.Cells[7,i])
           else a[i]:=0;
            summ:=0;
             for i:=1 to 10 do
             summ:=summ+a[i];
   tabl.Cells[7,11]:=FloatToStr(summ);//,ffGeneral,6,5);
tabl.Cells[10,1]:=floattostr((strtofloat(tabl.Cells[1,1]))*(strtofloat(tabl.Cells
[9,1])));

tabl.Cells[10,2]:=floattostr((strtofloat(tabl.Cells[1,2]))*(strtofloat(tabl.Cells
[9,2])));
tabl.Cells[10,3]:=floattostr((strtofloat(tabl.Cells[1,3]))*(strtofloat(tabl.Cells
[9,3])));
tabl.Cells[10,4]:=floattostr((strtofloat(tabl.Cells[1,4]))*(strtofloat(tabl.Cells
[9,4])));
tabl.Cells[10,5]:=floattostr((strtofloat(tabl.Cells[1,5]))*(strtofloat(tabl.Cells
[9,5])));
tabl.Cells[10,6]:=floattostr((strtofloat(tabl.Cells[1,6]))*(strtofloat(tabl.Cells
[9,6])));
tabl.Cells[10,7]:=floattostr((strtofloat(tabl.Cells[1,7]))*(strtofloat(tabl.Cells
[9,7])));
tabl.Cells[10,8]:=floattostr((strtofloat(tabl.Cells[1,8]))*(strtofloat(tabl.Cells
[9,8])));
tabl.Cells[10,9]:=floattostr((strtofloat(tabl.Cells[1,9]))*(strtofloat(tabl.Cells
[9,9])));
```

```
tabl.Cells[10,10]:=floattostr((strtofloat(tabl.Cells[1,10]))*(strtofloat(tabl.Ce
lls[9,10])))
  for i:=1 to tabl.RowCount do
  for j:=1 to 2 do
  b:=StrToFloat(tabl.Cells[1,i])-StrToFloat(tabl.Cells[2,i]);
      tabl.Cells[3,i]:=FloatToStr(b); }
      for i:=1 to 10 do
       if tabl.Cells[10,i]<>''
        then
         a[i]:=StrToFloat(tabl.Cells[10,i])
          else a[i]:=0;
           summ:=0;
            for i:=1 to 10 do
            summ:=summ+a[i];
   tabl.Cells[10,11]:=FloatToStr(summ);//,ffGeneral,7,6);
tabl.Cells[11,1]:=floattostr((strtofloat(tabl.Cells[3,1]))*(strtofloat(tabl.Cells
[9,1])))
tabl.Cells[11,2]:=floattostr((strtofloat(tabl.Cells[3,2]))*(strtofloat(tabl.Cells
[9,2])));
tabl.Cells[11,3]:=floattostr((strtofloat(tabl.Cells[3,3]))*(strtofloat(tabl.Cells
[9,3])));/
tabl.Cells[11,4]:=floattostr((strtofloat(tabl.Cells[3,4]))*(strtofloat(tabl.Cells
[9,4])));
tabl.Cells[11,5]:=floattostr((strtofloat(tabl.Cells[3,5]))*(strtofloat(tabl.Cells
[9,5])));
tabl.Cells[11,6]:=floattostr((strtofloat(tabl.Cells[3,6]))*(strtofloat(tabl.Cells
[9,6])));
tabl.Cells[11,7]:=floattostr((strtofloat(tabl.Cells[3,7]))*(strtofloat(tabl.Cells
[9,7])));/
tabl.Cells[11,8]:=floattostr((strtofloat(tabl.Cells[3,8]))*(strtofloat(tabl.Cells
[9,8])));
tabl.Cells[11,9]:=floattostr((strtofloat(tabl.Cells[3,9]))*(strtofloat(tabl.Cells
[9,9])));
tabl.Cells[11,10]:=floattostr((strtofloat(tabl.Cells[3,10]))*(strtofloat(tabl.Ce
lls[9,10])));
 { for i:=1 to tabl.RowCount do
  for j:=1 to 2 do
  b:=StrToFloat(tabl.Cells[1,i])-StrToFloat(tabl.Cells[2,i]);
      tabl.Cells[3,i]:=FloatToStr(b); }
      for i:=1 to 10 do
```



```
       if tabl.Cells[11,i]<>''
        then
         a[i]:=StrToFloat(tabl.Cells[11,i])
          else a[i]:=0;
           summ:=0;
            for i:=1 to 10 do
             summ:=summ+a[i];
   tabl.Cells[11,11]:=FloatToStr(summ);//,ffGeneral,7,6);
tabl.Cells[12,1]:=floattostr((strtofloat(tabl.Cells[4,1]))*(strtofloat(tabl.Cells
[9,1])));
tabl.Cells[12,2]:=floattostr((strtofloat(tabl.Cells[4,2]))*(strtofloat(tabl.Cells
[9,2])));
tabl.Cells[12,3]:=floattostr((strtofloat(tabl.Cells[4,3]))*(strtofloat(tabl.Cells
[9,3])));
tabl.Cells[12,4]:=floattostr((strtofloat(tabl.Cells[4,4]))*(strtofloat(tabl.Cells
[9,4])));

tabl.Cells[12,5]:=floattostr((strtofloat(tabl.Cells[4,5]))*(strtofloat(tabl.Cells
[9,5])));;
tabl.Cells[12,6]:=floattostr((strtofloat(tabl.Cells[4,6]))*(strtofloat(tabl.Cells
[9,6])));
tabl.Cells[12,7]:=floattostr((strtofloat(tabl.Cells[4,7]))*(strtofloat(tabl.Cells
[9,7])));/
tabl.Cells[12,8]:=floattostr((strtofloat(tabl.Cells[4,8]))*(strtofloat(tabl.Cells
[9,8])));
tabl.Cells[12,9]:=floattostr((strtofloat(tabl.Cells[4,9]))*(strtofloat(tabl.Cells
[9,9])));
tabl.Cells[12,10]:=floattostr((strtofloat(tabl.Cells[4,10]))*(strtofloat(tabl.Ce
lls[9,10])))
 { for i:=1 to tabl.RowCount do
  for j:=1 to 2 do
  b:=StrToFloat(tabl.Cells[1,i])-StrToFloat(tabl.Cells[2,i]);
      tabl.Cells[3,i]:=FloatToStr(b); }
      for i:=1 to 10 do
       if tabl.Cells[12,i]<>''
        then
         a[i]:=StrToFloat(tabl.Cells[12,i])
          else a[i]:=0;
           summ:=0;
            for i:=1 to 10 do
```

```
       summm:=summ+a[i];
   tabl.Cells[12,11]:=FloatToStr(summ);//,ffGeneral,6,5);
  o:=StrToFloat(tabl.Cells[7,11]);
  n:=StrToFloat(tabl.Cells[6,11]);
  m:=StrToFloat(tabl.Cells[5,11]);
  l:=StrToFloat(tabl.Cells[4,11]);
  k:=StrToFloat(tabl.Cells[3,11]);
  g:=StrToFloat(tabl.Cells[8,11]);
  j:=StrToFloat(tabl.Cells[2,11]);
  r:=StrToFloat(tabl.Cells[12,11]);
  q:=StrToFloat(tabl.Cells[11,11]);
  p:=StrToFloat(tabl.Cells[10,11]);
  h:=StrToFloat(tabl.Cells[9,11]);
  c:=StrToFloat(tabl.Cells[1,11]);
    n21:=(o*(m*k*j+l*g*k+k*l*g-k*k*k-l*l*j-m*g*g))-
(n*(n*k*j+l*g*l+k*m*g-k*k*l-l*m*j-n*g*g))+  //d
       (m*(n*l*j+m*g*l+k*m*k-k*l*l-m*m*j-n*g*k))-
(l*(n*l*g+m*k*l+l*m*k-l*l*l-m*m*g-n*k*k));
    n28:=r*(m*k*j+l*c*k+k*l*c-k*k*k-l*l*j-m*c*c)-
n*(q*k*j+l*c*h+k*p*c-k*k*h-l*p*j-q*c*c)+
       m*(q*l*j+m*c*h+k*p*k-k*l*h-m*p*j-q*c*k)-
l*(q*l*c+m*k*h+l*p*k-l*l*h-m*p*c-q*k*k);
    n35:=o*(q*k*j+l*g*h+k*p*g-k*k*h-l*p*j-q*g*g)-
r*(n*k*j+l*g*l+k*m*g-k*k*l-l*m*j-n*g*g)+  //d2
       m*(n*p*j+q*g*l+k*m*h-k*p*l-q*m*j-n*g*h)-
l*(n*p*g+q*k*l+l*m*h-l*p*l-q*m*g-n*k*h);
    n42:=o*(m*p*j+q*g*k+k*l*h-k*p*k-q*l*j-m*g*h)-
n*(n*p*j+q*g*l+k*m*h-k*p*l-q*m*j-n*g*h)+  //d3
       r*(n*l*j+m*g*l+k*m*k-k*l*l-m*m*j-n*g*k)-
l*(n*l*h+m*p*l+q*m*k-q*l*l-m*m*h-n*p*k);
    n49:=o*(m*k*h+l*p*k+q*l*g-q*k*k-l*l*h-m*p*g)-
n*(n*k*h+l*p*l+q*m*g-q*k*l-l*m*h-n*p*g)+  //d4
       m*(n*l*h+m*p*l+q*m*k-q*l*l-m*m*h-n*p*k)-
r*(n*l*g+m*k*l+l*m*k-l*l*l-m*m*g-n*k*k);
    if (n21=0) or (n28=0) or (n35=0) or (n42=0) or (n49=0) then
      MessageDlg('Визначник матриці рівний
нулю.',mtInformation,[mbOK],0);  // Система або несумісна або має
безліч розв"язків.
      try
    d:=n28/n21;     //a
```



```
d1:=n35/n21;    //b
    d2:=n42/n21;    //c
    d3:=n49/n21;    //d
     except
      on EZeroDivide do MessageDlg('Ділення на
нуль!',mtError,[mbOK],0);
     end;
    label1.Caption:=FloatToStrf((d),ffGeneral,6,5);
    label2.Caption:=FloatToStrf((d1),ffGeneral,6,5);
    label3.Caption:=FloatToStrf((d2),ffGeneral,6,6);
    label4.Caption:=FloatToStrf((d3),ffGeneral,6,6);
tabl.Cells[13,1]:=floattostr(d*(strtofloat(tabl.Cells[4,1]))+d1*(strtofloat(tabl
.Cells[3,1]))+ d2*(strtofloat(tabl.Cells[1,1]))+d3);//,ffGeneral,6,5);
tabl.Cells[13,2]:=floattostr(d*(strtofloat(tabl.Cells[4,2]))+d1*(strtofloat(tabl
.Cells[3,2]))+d2*(strtofloat(tabl.Cells[1,2]))+d3);//,ffGeneral,6,5);
tabl.Cells[13,3]:=floattostr(d*(strtofloat(tabl.Cells[4,3]))+d1*(strtofloat(tabl
.Cells[3,3]))+d2*(strtofloat(tabl.Cells[1,3]))+d3);//,ffGeneral,6,5);
tabl.Cells[13,4]:=floattostr(d*(strtofloat(tabl.Cells[4,4]))+d1*(strtofloat(tabl
.Cells[3,4]))+d2*(strtofloat(tabl.Cells[1,4]))+d3);//,ffGeneral,6,5);
tabl.Cells[13,5]:=floattostr(d*(strtofloat(tabl.Cells[4,5]))+d1*(strtofloat(tabl
.Cells[3,5]))+ d2*(strtofloat(tabl.Cells[1,5]))+d3);//,ffGeneral,6,5);
tabl.Cells[13,6]:=floattostr(d*(strtofloat(tabl.Cells[4,6]))+d1*(strtofloat(tabl
.Cells[3,6]))+d2*(strtofloat(tabl.Cells[1,6]))+d3);//,ffGeneral,6,5);
tabl.Cells[13,7]:=floattostr(d*(strtofloat(tabl.Cells[4,7]))+d1*(strtofloat(tabl
.Cells[3,7]))+d2*(strtofloat(tabl.Cells[1,7]))+d3);//,ffGeneral,6,
tabl.Cells[13,8]:=floattostr(d*(strtofloat(tabl.Cells[4,8]))+d1*(strtofloat(tabl
.Cells[3,8]))+d2*(strtofloat(tabl.Cells[1,8]))+d3);//,ffGeneral,6,5);
tabl.Cells[13,9]:=floattostr(d*(strtofloat(tabl.Cells[4,9]))+d1*(strtofloat(tabl
.Cells[3,9]))+  d2*(strtofloat(tabl.Cells[1,9]))+d3);//,ffGeneral,6,5);
tabl.Cells[13,10]:=floattostr(d*(strtofloat(tabl.Cells[4,10]))+d1*(strtofloat(t
abl.Cells[3,10])) d2*(strtofloat(tabl.Cells[1,10]))+d3);//,ffGeneral,6,5);
      { for i:=1 to tabl.RowCount do
  for j:=1 to 2 do
  b:=StrToFloat(tabl.Cells[1,i])-StrToFloat(tabl.Cells[2,i]);
     tabl.Cells[3,i]:=FloatToStr(b); }
     for i:=1 to 10 do
      if tabl.Cells[13,i]<>''
       then
        a[i]:=StrToFloat(tabl.Cells[13,i])
         else a[i]:=0;
```

```
summm:=0;
            for i:=1 to 10 do
            summm:=summm+a[i];
   tabl.Cells[13,11]:=FloatToStr(summ);//,ffGeneral,6,5);
  tabl.Cells[14,1]:=floattostr((strtofloat(tabl.Cells[9,1]))-
(strtofloat(tabl.Cells[13,1])));//,ffGeneral,6,5);
  tabl.Cells[14,2]:=floattostr((strtofloat(tabl.Cells[9,2]))-
(strtofloat(tabl.Cells[13,2])));//,ffGeneral,6,5);
  tabl.Cells[14,3]:=floattostr((strtofloat(tabl.Cells[9,3]))-
(strtofloat(tabl.Cells[13,3])));//,ffGeneral,6,5);
  tabl.Cells[14,4]:=floattostr((strtofloat(tabl.Cells[9,4]))-
(strtofloat(tabl.Cells[13,4])));//,ffGeneral,6,5);
  tabl.Cells[14,5]:=floattostr((strtofloat(tabl.Cells[9,5]))-
(strtofloat(tabl.Cells[13,5])));//,ffGeneral,6,5);
  tabl.Cells[14,6]:=floattostr((strtofloat(tabl.Cells[9,6]))-
(strtofloat(tabl.Cells[13,6])));//,ffGeneral,6,5);
  tabl.Cells[14,7]:=floattostr((strtofloat(tabl.Cells[9,7]))-
(strtofloat(tabl.Cells[13,7])));//,ffGeneral,6,5);
  tabl.Cells[14,8]:=floattostr((strtofloat(tabl.Cells[9,8]))-
(strtofloat(tabl.Cells[13,8])));//,ffGeneral,6,5);
  tabl.Cells[14,9]:=floattostr((strtofloat(tabl.Cells[9,9]))-
(strtofloat(tabl.Cells[13,9])));//,ffGeneral,6,5);
  tabl.Cells[14,10]:=floattostr((strtofloat(tabl.Cells[9,10]))-
(strtofloat(tabl.Cells[13,10])));//,ffGeneral,6,5);
 { for i:=1 to tabl.RowCount do
  for j:=1 to 2 do
  b:=StrToFloat(tabl.Cells[1,i])-StrToFloat(tabl.Cells[2,i]);
       tabl.Cells[3,i]:=FloatToStr(b); }
       for i:=1 to 10 do
        if tabl.Cells[14,i]<>''
         then
          a[i]:=StrToFloat(tabl.Cells[14,i])
           else a[i]:=0;
            summm:=0;
             for i:=1 to 10 do
              summm:=summm+a[i];
   tabl.Cells[14,11]:=FloatToStr(summm);//,ffGeneral,6,5);
  tabl.Cells[15,1]:=floattostr(sqr(strtofloat(tabl.Cells[14,1])));
  tabl.Cells[15,2]:=floattostr(sqr(strtofloat(tabl.Cells[14,2])));
  tabl.Cells[15,3]:=floattostr(sqr(strtofloat(tabl.Cells[14,3])));;
```



```
      tabl.Cells[15,4]:=floattostr(sqr(strtofloat(tabl.Cells[14,4])));
tabl.Cells[15,5]:=floattostr(sqr(strtofloat(tabl.Cells[14,5])));
tabl.Cells[15,6]:=floattostr(sqr(strtofloat(tabl.Cells[14,6])))
  tabl.Cells[15,7]:=floattostr(sqr(strtofloat(tabl.Cells[14,7])));
tabl.Cells[15,8]:=floattostr(sqr(strtofloat(tabl.Cells[14,8])));/
tabl.Cells[15,9]:=floattostr(sqr(strtofloat(tabl.Cells[14,9])));
tabl.Cells[15,10]:=floattostr(sqr(strtofloat(tabl.Cells[14,10]))
       for i:=1 to 10 do
        if tabl.Cells[15,i]<>''
         then
          a[i]:=StrToFloat(tabl.Cells[15,i])
           else a[i]:=0;
            summm:=0;
             for i:=1 to 10 do
              summm:=summm+a[i];
   tabl.Cells[15,11]:=FloatToStr(summm);//,ffGeneral,6,5);
tabl.Cells[16,1]:=floattostr(sqr(strtofloat(tabl.Cells[9,1]))));
  tabl.Cells[16,2]:=floattostr(sqr(strtofloat(tabl.Cells[9,2]))));
  tabl.Cells[16,3]:=floattostr(sqr(strtofloat(tabl.Cells[9,3]))));
  tabl.Cells[16,4]:=floattostr(sqr(strtofloat(tabl.Cells[9,4]))));
  tabl.Cells[16,5]:=floattostr(sqr(strtofloat(tabl.Cells[9,5]))));
  tabl.Cells[16,6]:=floattostr(sqr(strtofloat(tabl.Cells[9,6]))));
  tabl.Cells[16,7]:=floattostr(sqr(strtofloat(tabl.Cells[9,7]))));
  tabl.Cells[16,8]:=floattostr(sqr(strtofloat(tabl.Cells[9,8]))));
tabl.Cells[16,9]:=floattostr(sqr(strtofloat(tabl.Cells[9,9]))));
tabl.Cells[16,10]:=floattostr(sqr(strtofloat(tabl.Cells[9,10]))));
       for i:=1 to 10 do
        if tabl.Cells[16,i]<>''
         then
          a[i]:=StrToFloat(tabl.Cells[16,i])
           else a[i]:=0;
            summm:=0;
             for i:=1 to 10 do
              summm:=summm+a[i];
   tabl.Cells[16,11]:=FloatToStr(summm);//,ffGeneral,6,5);
  a44:=o*(m*k-l*l)+n*(l*m-n*k)+m*(n*l-m*m);//o*m*k+n*l*m+m*n*l-
m*m*m-n*n*k-o*l*l;
  a22:=o*(k*j-g*g)+m*(g*l-m*j)+l*(m*g-k*l);//o*k*j+m*g*l+l*m*g-
l*k*l-m*m*j-o*g*g;
```

```
a33:=o*(m*j-k*k)+n*(k*l-n*j)+l*(n*k-m*l);//o*m*j+n*k*l+l*n*k-l*m*l-
n*n*j-o*k*k;
  a11:=m*(k*j-g*g)+l*(g*k-l*j)+k*(l*g-k*k);//m*k*j+l*g*k+k*l*g-k*k*k-
l*l*j-m*g*g;
  h54:=(strtofloat(tabl.Cells[16,11]))-(d*(strtofloat(tabl.Cells[12,11])))-
(d1*(strtofloat(tabl.Cells[11,11])))-
      (d2*(strtofloat(tabl.cells[10,11])))-(d3*(strtofloat(tabl.cells[9,11])));
  h56:=h54-(strtofloat(tabl.cells[15,11]));
  e:=StrtoFloat(tabl.Cells[15,11]);
  label6.Caption:='A11='+#32+floattostrf((a11),ffGeneral,6,5);
  label7.Caption:='A22='+#32+floattostrf((a22),ffGeneral,6,5);
  label8.Caption:='A33='+#32+floattostrf((a33),ffGeneral,7,6);
  label9.Caption:='A44='+#32+floattostrf((a44),ffGeneral,7,6);

label5.Caption:=floattostrf((h54),ffGeneral,6,5)+#13+floattostrf((e),ffGener
al,6,5)+#13+floattostrf((h56),ffGeneral,6,5);
  mv:=sqrt((strtofloat(tabl.Cells[15,11]))/6);
  ma:=mv*(sqrt(a11/n21));
  mb:=mv*(sqrt(a22/n21));
  mc:=mv*(sqrt(a33/n21));
  md:=mv*(sqrt(a44/n21));
  zn1:=d/ma;
  zn2:=d1/mb;
  zn3:=d2/mc;
  zn4:=d3/md;
  label10.Caption:='Середня кв.похибка
ваги='+#32+floattostrf((mv),ffGeneral,6,5);
  label11.Caption:='Середня кв.похибка ma=';
  label12.Caption:='Середня кв.похибка mb=';
  label13.Caption:='Середня кв.похибка mc=';
  label14.Caption:='Середня кв.похибка md=';
  label22.Caption:=floattostrf((ma),ffGeneral,7,6);
  label23.Caption:=floattostrf((mb),ffGeneral,7,6);
  label24.Caption:=floattostrf((mc),ffGeneral,7,6);
  label25.Caption:=floattostrf((md),ffGeneral,7,6);
    //vagi
  s31:=StrToFloat(tabl.Cells[4,1]); s32:=StrToFloat(tabl.Cells[4,2]);
  s33:=StrToFloat(tabl.Cells[4,3]); s34:=StrToFloat(tabl.Cells[4,4]);
  s35:=StrToFloat(tabl.Cells[4,5]); s36:=StrToFloat(tabl.Cells[4,6]);
  s37:=StrToFloat(tabl.Cells[4,7]); s38:=StrToFloat(tabl.Cells[4,8]);
```



```
s39:=StrToFloat(tabl.Cells[4,9]); s3:=StrToFloat(tabl.Cells[4,10]);
   s21:=StrToFloat(tabl.Cells[3,1]); s22:=StrToFloat(tabl.Cells[3,2]);
   s23:=StrToFloat(tabl.Cells[3,3]); s24:=StrToFloat(tabl.Cells[3,4]);
   s25:=StrToFloat(tabl.Cells[3,5]); s26:=StrToFloat(tabl.Cells[3,6]);
   s27:=StrToFloat(tabl.Cells[3,7]); s28:=StrToFloat(tabl.Cells[3,8]);
   s29:=StrToFloat(tabl.Cells[3,9]); s2:=StrToFloat(tabl.Cells[3,10]);
    ss1:=StrToFloat(tabl.Cells[1,1]); ss2:=StrToFloat(tabl.Cells[1,2]);
    ss3:=StrToFloat(tabl.Cells[1,3]); ss4:=StrToFloat(tabl.Cells[1,4]);
    ss5:=StrToFloat(tabl.Cells[1,5]); ss6:=StrToFloat(tabl.Cells[1,6]);
    ss7:=StrToFloat(tabl.Cells[1,7]); ss8:=StrToFloat(tabl.Cells[1,8]);
    ss9:=StrToFloat(tabl.Cells[1,9]); ss:=StrToFloat(tabl.Cells[1,10]);
     s01:=StrToFloat(tabl.Cells[2,1]); s02:=StrToFloat(tabl.Cells[2,2]);
     s03:=StrToFloat(tabl.Cells[2,3]); s04:=StrToFloat(tabl.Cells[2,4]);
     s05:=StrToFloat(tabl.Cells[2,5]); s06:=StrToFloat(tabl.Cells[2,6]);
     s07:=StrToFloat(tabl.Cells[2,7]); s08:=StrToFloat(tabl.Cells[2,8]);
     s09:=StrToFloat(tabl.Cells[2,9]); s0:=StrToFloat(tabl.Cells[2,10]);
  x1:=(m*k*j+l*g*k+k*l*g-k*k*k-l*l*j-m*g*g)/n21;
  x2:=(n*k*j+l*g*l+k*m*g-k*k*l-l*m*j-n*g*g)/(-1*n21);
  x3:=(n*l*j+m*g*l+k*m*k-k*l*l-m*m*j-n*g*k)/n21;
  x4:=(n*l*g+m*k*l+l*m*k-l*l*l-m*m*g-n*k*k)/(-1*n21);
  x5:=(n*k*j+m*g*k+l*l*g-l*k*k-m*l*j-n*g*g)/(-1*n21);
  x6:=(o*k*j+m*g*l+l*m*g-l*k*l-m*m*j-o*g*g)/n21;
  x7:=(o*l*j+n*g*l+l*m*k-l*l*l-n*m*j-o*g*k)/(-1*n21);
  x8:=(o*l*g+n*k*l+m*m*k-m*l*l-n*m*g-o*k*k)/n21;
  x9:=(n*l*j+m*k*k+l*m*g-l*l*k-m*m*j-n*k*g)/n21;
  x10:=(o*l*j+m*k*l+l*n*g-l*l*l-m*n*j-o*k*g)/(-1*n21);
  x11:=(o*m*j+n*k*l+l*n*k-l*m*l-n*n*j-o*k*k)/n21;
  x12:=(o*m*g+n*l*l+m*n*k-m*m*l-n*n*g-o*l*k)/(-1*n21);
  x13:=(n*l*g+m*k*l+l*m*k-l*l*l-m*m*g-n*k*k)/(-1*n21);
  x14:=(o*l*g+m*k*m+l*n*k-l*l*m-m*n*g-o*k*k)/n21;
  x15:=(o*m*g+n*k*m+l*n*l-l*m*m-n*n*g-o*k*l)/(-1*n21);
  x16:=(o*m*k+n*l*m+m*n*l-m*m*m-n*n*k-o*l*l)/n21;
y1:=s31*(s31*x1+s21*x5+ss1*x9+s01*x13)+s21*(s31*x2+s21*x6+ss1*x1
0+s01*x14)+
ss1*(s31*x3+s21*x7+ss1*x11+s01*x15)+s01*(s31*x4+s21*x8+ss1*x12+s
01*x16);
y2:=s32*(s32*x1+s22*x5+ss2*x9+s02*x13)+s22*(s32*x2+s22*x6+ss2*x1
0+s02*x14)+
ss2*(s32*x3+s22*x7+ss2*x11+s02*x15)+s02*(s32*x4+s22*x8+ss2*x12+s
02*
```

```
x16);
y3:=s33*(s33*x1+s23*x5+ss3*x9+s03*x13)+s23*(s33*x2+s23*x6+ss3*x1
0+s03*x14)+ss3*(s33*x3+s23*x7+ss3*x11+s03*x15)+s03*(s33*x4+s23*
x8+ss3*x12+s03*x16);
y4:=s34*(s34*x1+s24*x5+ss4*x9+s04*x13)+s24*(s34*x2+s24*x6+ss4*x1
0+s04*x14)+
ss4*(s34*x3+s24*x7+ss4*x11+s04*x15)+s04*(s34*x4+s24*x8+ss4*x12+s
04*x16);
y5:=s35*(s35*x1+s25*x5+ss5*x9+s05*x13)+s25*(s35*x2+s25*x6+ss5*x1
0+s05*x14)+
ss5*(s35*x3+s25*x7+ss5*x11+s05*x15)+s05*(s35*x4+s25*x8+ss5*x12+s
05*x16);
y6:=s36*(s36*x1+s26*x5+ss6*x9+s06*x13)+s26*(s36*x2+s26*x6+ss6*x1
0+s06*x14)+
ss6*(s36*x3+s26*x7+ss6*x11+s06*x15)+s06*(s36*x4+s26*x8+ss6*x12+s
06*x16);
y7:=s37*(s37*x1+s27*x5+ss7*x9+s07*x13)+s27*(s37*x2+s27*x6+ss7*x1
0+s07*x14)+
ss7*(s37*x3+s27*x7+ss7*x11+s07*x15)+s07*(s37*x4+s27*x8+ss7*x12+s
07*x16);
y8:=s38*(s38*x1+s28*x5+ss8*x9+s08*x13)+s28*(s38*x2+s28*x6+ss8*x1
0+s08*x14)+
ss8*(s38*x3+s28*x7+ss8*x11+s08*x15)+s08*(s38*x4+s28*x8+ss8*x12+s
08*x16);
y9:=s39*(s39*x1+s29*x5+ss9*x9+s09*x13)+s29*(s39*x2+s29*x6+ss9*x1
0+s09*x14)+
ss9*(s39*x3+s29*x7+ss9*x11+s09*x15)+s09*(s39*x4+s29*x8+ss9*x12+s
09*x16);
y10:=s3*(s3*x1+s2*x5+ss*x9+s0*x13)+s2*(s3*x2+s2*x6+ss*x10+s0*x14
)+
   ss*(s3*x3+s2*x7+ss*x11+s0*x15)+s0*(s3*x4+s2*x8+ss*x12+s0*x16);
 tabl.Cells[18,1]:=FloatToStr(y1);
 tabl.Cells[18,2]:=FloatToStr(y2);
 tabl.Cells[18,3]:=FloatToStr(y3);
 tabl.Cells[18,4]:=FloatToStr(y4);
 tabl.Cells[18,5]:=FloatToStr(y5);
 tabl.Cells[18,6]:=FloatToStr(y6);
 tabl.Cells[18,7]:=FloatToStr(y7);
 tabl.Cells[18,8]:=FloatToStr(y8);
 tabl.Cells[18,9]:=FloatToStr(y9);
```



```
 tabl.Cells[18,10]:=FloatToStr(y10);
  tabl.Cells[19,1]:=FloatToStr(sqrt(y1));
  tabl.Cells[19,2]:=FloatToStr(sqrt(y2));
  tabl.Cells[19,3]:=FloatToStr(sqrt(y3));
  tabl.Cells[19,4]:=FloatToStr(sqrt(y4));
  tabl.Cells[19,5]:=FloatToStr(sqrt(y5));
  tabl.Cells[19,6]:=FloatToStr(sqrt(y6));
  tabl.Cells[19,7]:=FloatToStr(sqrt(y7));
  tabl.Cells[19,8]:=FloatToStr(sqrt(y8));
  tabl.Cells[19,9]:=FloatToStr(sqrt(y9));
  tabl.Cells[19,10]:=FloatToStr(sqrt(y10));
   tabl.Cells[20,1]:=FloatToStr(mv*strtofloat(tabl.Cells[19,1]));
   tabl.Cells[20,2]:=FloatToStr(mv*strtofloat(tabl.Cells[19,2]));
   tabl.Cells[20,3]:=FloatToStr(mv*strtofloat(tabl.Cells[19,3]));
   tabl.Cells[20,4]:=FloatToStr(mv*strtofloat(tabl.Cells[19,4]));
   tabl.Cells[20,5]:=FloatToStr(mv*strtofloat(tabl.Cells[19,5]));
   tabl.Cells[20,6]:=FloatToStr(mv*strtofloat(tabl.Cells[19,6]));
   tabl.Cells[20,7]:=FloatToStr(mv*strtofloat(tabl.Cells[19,7]));
   tabl.Cells[20,8]:=FloatToStr(mv*strtofloat(tabl.Cells[19,8]));
   tabl.Cells[20,9]:=FloatToStr(mv*strtofloat(tabl.Cells[19,9]));
   tabl.Cells[20,10]:=FloatToStr(mv*strtofloat(tabl.Cells[19,10]));

  BitBtn3.Enabled:=true;
 end;

procedure TFRozr.BitBtn2Click(Sender: TObject);
begin
FRozr.Close;
end;

procedure TFRozr.BitBtn3Click(Sender: TObject);
begin
 Panel2.Visible:=true;
 Panel1.Visible:=false;
 Panel3.Visible:=false;
 Panel2.Align:=alClient;
 FRozr.Caption:='Графіки';
end;

procedure TFRozr.BitBtn4Click(Sender: TObject);
```

```
var
 j:integer;
begin
  Chart1.Height:=650;
  Chart1.Visible:=true;
  Chart2.Visible:=false;
  Chart3.Visible:=false;
  Chart4.Visible:=false;
  Chart5.Visible:=false;
  Chart6.Visible:=false;
  Chart7.Visible:=false;
  series1.Clear;
  Chart1.Title.Text.Clear;
 Chart1.LeftAxis.Title.Caption:='Попит на товар, y';
 Chart1.BottomAxis.Title.Caption:='Ціна на товар, x';
 Chart1.Title.Text.Add('ІСТИННА МОДЕЛЬ');
 for j:=1 to 10 do
 Series1.AddXY(strtofloat(tabl.Cells[8,j]),strtofloat(tabl.Cells[9,j]),'',clre;
end;

procedure TFRozr.BitBtn5Click(Sender: TObject);
   var
 j:integer;
begin
  Chart2.Height:=650;
  Chart1.Visible:=false;
  Chart2.Visible:=true;
  Chart3.Visible:=false;
  Chart4.Visible:=false;
  Chart5.Visible:=false;
  Chart6.Visible:=false;
  Chart7.Visible:=false;
  series2.Clear;
 Chart2.Title.Text.Clear;
 Chart2.LeftAxis.Title.Caption:='Попит на товар, y';
 Chart2.BottomAxis.Title.Caption:='Ціна на товар, x';
 Chart2.Title.Text.Add('СПОТВОРЕНА МОДЕЛЬ');
 for j:=1 to 10 do
 Series2.AddXY(strtofloat(tabl.Cells[1,j]),strtofloat(tabl.Cells[9,j]),'',clre;
end;
```



```
procedure TFRozr.BitBtn6Click(Sender: TObject);
var
 i,j:integer;
begin
  Chart3.Height:=650;
  Chart1.Visible:=false;
  Chart2.Visible:=false;
  Chart4.Visible:=false;
  Chart5.Visible:=false;
  Chart6.Visible:=false;
  Chart7.Visible:=false;
  Chart3.Visible:=true;
  series3.Clear;
  Series4.Clear;
 Chart3.Title.Text.Clear;
 Chart3.LeftAxis.Title.Caption:='Значення похибок';
 Chart3.BottomAxis.Title.Caption:='Номера спостережень';
 Chart3.Title.Text.Add('ІСТИННІ І АБСОЛЮТНІ ПОХИБКИ');
 Series3.Title:='істинні похибки';
 Series4.Title:='абсолютні похибки';
 for j:=1 to 10 do

Series3.AddXY(strtofloat(tabl.Cells[0,j]),strtofloat(tabl.Cells[17,j]),'',clred);
 for j:=1 to 10 do

Series4.AddXY(strtofloat(tabl.Cells[0,j]),strtofloat(tabl.Cells[14,j]),'',clgree
n);
end;
procedure TFRozr.BitBtn7Click(Sender: TObject);
var
 j:integer;
begin
  Chart4.Height:=650;
  Chart1.Visible:=false;
  Chart2.Visible:=false;
  Chart3.Visible:=false;
  Chart5.Visible:=false;
  Chart6.Visible:=false;
  Chart7.Visible:=false;
  Chart4.Visible:=true;
```

```
 Series5.Clear;
 Chart4.Title.Text.Clear;
 Chart4.LeftAxis.Title.Caption:='Значення обернених ваг';
 Chart4.BottomAxis.Title.Caption:='Номера спостережень';
 Chart4.Title.Text.Add('ОБЕРНЕНІ ВАГИ ФУНКЦІЇ');
 for j:=1 to 10 do
Series5.AddXY(strtofloat(tabl.Cells[0,j]),strtofloat(tabl.Cells[18,j]),'',clred);
end;
procedure TFRozr.BitBtn8Click(Sender: TObject);
var
 j:integer;
begin
  Chart5.Height:=650;
  Chart1.Visible:=false;
  Chart2.Visible:=false;
  Chart3.Visible:=false;
  Chart4.Visible:=false;
  Chart6.Visible:=false;
  Chart7.Visible:=false;
  Chart5.Visible:=true;
  Series6.Clear;
 Chart5.Title.Text.Clear;
 Chart5.LeftAxis.Title.Caption:='Значення похибок';
 Chart5.BottomAxis.Title.Caption:='Номера спостережень';
 Chart5.Title.Text.Add('СЕРЕДНІ КВАДРАТИЧНІ ПОХИБКИ
ФУНКЦІЇ');
 for j:=1 to 10 do

Series6.AddXY(strtofloat(tabl.Cells[0,j]),strtofloat(tabl.Cells[20,j]),'',clred);
end;
procedure TFRozr.BitBtn9Click(Sender: TObject);
var
 j:integer;
begin
  Chart6.Height:=650;
  Chart1.Visible:=false;
  Chart2.Visible:=false;
  Chart3.Visible:=false;
  Chart4.Visible:=false;
  Chart5.Visible:=false;
```



```
Chart7.Visible:=false;
  Chart6.Visible:=true;
  series7.Clear;
  Series8.Clear;
 Chart6.Title.Text.Clear;
 Chart6.LeftAxis.Title.Caption:='Значення похибок';
 Chart6.BottomAxis.Title.Caption:='Номера спостережень';
 Chart6.Title.Text.Add('АБСОЛЮТНІ І СЕРЕДНІ КВАДРАТИЧНІ
ПОХИБКИ ФУНКЦІЙ');
 Series7.Title:='абсолютні похибки';
 Series8.Title:='середні кв.похибки';
 for j:=1 to 10 do
Series7.AddXY(strtofloat(tabl.Cells[0,j]),strtofloat(tabl.Cells[14,j]),'',clred);
 for j:=1 to 10 do
Series8.AddXY(strtofloat(tabl.Cells[0,j]),strtofloat(tabl.Cells[20,j]),'',clgree
);

end;
procedure TFRozr.BitBtn10Click(Sender: TObject);
var
 i,j:integer;
 a:array[1..4] of real;  zn1,zn2,zn3,zn4:real;
 b:array[1..4] of string;
 c:array[1..4] of real;
begin
   a[1]:=strtofloat(label22.Caption);
   a[2]:=strtofloat(label23.Caption);
   a[3]:=strtofloat(label24.Caption);
   a[4]:=strtofloat(label25.Caption);
   c[1]:=abs(strtofloat(label1.Caption)/strtofloat(label22.Caption));
   c[2]:=abs(strtofloat(label2.Caption)/strtofloat(label23.Caption));
   c[3]:=abs(strtofloat(label3.Caption)/strtofloat(label24.Caption));
   c[4]:=abs(strtofloat(label4.Caption)/strtofloat(label25.Caption));
   b[1]:='a';b[2]:='b';b[3]:='c';b[4]:='d';
  Chart7.Height:=655;
  Chart1.Visible:=false;
  Chart2.Visible:=false;
  Chart3.Visible:=false;
  Chart4.Visible:=false;
  Chart5.Visible:=false;
```

```
Chart6.Visible:=false;
  Chart7.Visible:=true;
  series9.Clear;
  series10.Clear;
  Series8.Clear;
 Chart7.Title.Text.Clear;
 Chart7.LeftAxis.Title.Caption:='Значення похибок';
 Chart7.BottomAxis.Title.Caption:='Назви коефіцієнтів';
 Chart7.Title.Text.Add('СЕРЕДНІ КВ. ПОХИБКИ КОЕФІЦІЄНТІВ І ЇХ
ЗНАЧУЩІСТЬ');
 Series9.Title:='середні кв. пох. коеф.';
 Series10.Title:='значущість коеф.';
  for i:=1 to 4 do
 Series9.Add(a[i],b[i],clred);
 for i:=1 to 4 do
 Series10.Add(c[i],b[i],clgreen);
end;

procedure TFRozr.BitBtn11Click(Sender: TObject);
begin
 Panel2.Visible:=false;
 Panel1.Visible:=true;
end;

procedure TFRozr.BitBtn12Click(Sender: TObject);
Var tmpH,tmpW,tmpWMargin,tmpHMargin:Longint;
    OldOrientation:TPrinterOrientation;
begin
  Screen.Cursor := crHourGlass;
  OldOrientation:=Printer.Orientation;
  Printer.Orientation:=poLandscape;
  try
    Printer.BeginDoc;
    try
      Printer.Title:='Графіки';

         Begin
             Chart1.PrintResolution:= -100;
             Chart2.PrintResolution:= -100;
             Chart3.PrintResolution:= -100;
```



```
             Chart4.PrintResolution:= -100;
             Chart5.PrintResolution:= -100;
             Chart6.PrintResolution:= -100;
             Chart7.PrintResolution:= -100;
          End;

      tmpW:=Printer.PageWidth;
      tmpWMargin:=Round(5.0*tmpW/100.0);
      tmpW:=tmpW-2*tmpWMargin;
      tmpW:=tmpW div 2;
      tmpH:=Printer.PageHeight;
      tmpHMargin:=Round(5.0*tmpH/100.0);
      tmpH:=tmpH-2*tmpHMargin;
      tmpH:=tmpH div 2;

      Chart1.PrintPartial( Rect( tmpWMargin,tmpHMargin,
                           tmpWMargin+tmpW,tmpHMargin+tmpH ) );
      Chart3.PrintPartial( Rect( tmpWMargin+tmpW,tmpHMargin,
                              tmpWMargin+2*tmpW,tmpHMargin+tmpH ) );
      Chart2.PrintPartial( Rect( tmpWMargin,tmpHMargin+tmpH,
                           tmpWMargin+tmpW,tmpHMargin+2*tmpH ) );
      Chart4.PrintPartial( Rect( tmpWMargin+tmpW,tmpHMargin+tmpH,
                           tmpWMargin+2*tmpW,tmpHMargin+2*tmpH ) );
      Chart5.PrintPartial( Rect( tmpWMargin+tmpW,tmpHMargin+tmpH,
                           tmpWMargin+tmpW,tmpHMargin+2*tmpH ) );
      Chart6.PrintPartial( Rect( tmpWMargin+tmpW,tmpHMargin+tmpH,
                           tmpWMargin+2*tmpW,tmpHMargin+2*tmpH ) );
      Chart7.PrintPartial( Rect( tmpWMargin+tmpW,tmpHMargin+tmpH,
                           tmpWMargin+tmpW,tmpHMargin+2*tmpH ) );
      Printer.EndDoc;
    except
      on Exception do
      Begin
        Printer.Abort;
        Printer.EndDoc;
        Raise;
      end;
    end;
  finally
    Printer.Orientation:=OldOrientation;
```

N.S.Pancheliuga

CONSTRUCTION AND RESEARCH OF ECONOMIC MATHEMATICAL MODEL OF DEPENDENCE OF DEMAND ON A COMMODITY FROM PRICE BY THE METHOD OF MONTE-CARLO

MODEL 81KIN-M42

On the basis of facts sheets of dependence of cost of commodity and demand on him the built mathematical model as cube to the polynomial on the method of leastsquares.

In-process this middle quadratic errors which over are brought to set rationed are generated, the disfigured model is built, is counterbalanced on the method of leastsquares. There are more credible values of coefficients a, b, c, d of cube to the polynomial of approximating mathematical model. Estimation of exactness is done and summarizing conclusions are given.

The applied method of statistical tests of Monte-Carlo enabled to conduct large-scale researches and to collect large statistics.

For students and graduate students of the Economic faculty and faculty of Cybernetics of IEGU.

Scientific leader:
R.M.Litnarovich, associate professor
candidate of engineerings sciences

Надія Степанівна Панчелюга,
магістр інформаційних технологій

ПОБУДОВА І ДОСЛІДЖЕННЯ ЕКОНОМІКО-МАТЕМАТИЧНОЇ МОДЕЛІ ЗАЛЕЖНОСТІ ЦІНИ ТОВАРУ І ПОПИТУ МЕТОДОМ СТАТИСТИЧНИХ ВИПРОБУВАНЬ МОНТЕ-КАРЛО

МОДЕЛЬ 81КІН-М42

НАУКОВЕ ВИДАННЯ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІЖНАРОДНИЙ ЕКОНОМІКО-ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
ІМЕНІ АКАДЕМІКА СТЕПАНА ДЕМ'ЯНЧУКА

ФАКУЛЬТЕТ КІБЕРНЕТИКИ

КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

Комп'ютерний набір, верстка, редагування
і макетування та дизайн в редакторі
Microsoft ® Office ® Word 2003
Н.С.Панчелюга

Науковий керівник: Р.М.Літнарович,
доцент, кандидат технічних наук

33027 Рівне , Україна
Вул..С.Дем'янчука, 4, корпус 1
Телефон : (+00380) 362 23 – 73 – 09
Факс :(+00380) 362 23 – 01 – 86
E-mail:mail@regi.rovno.ua