Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Кафедра прикладної математики

**Застосування методу Крігінга для побудови сурогатних моделей**

Курсова робота студента 3 курсу групи ПМп-32

Лабенського Данила

Науковий керівник: доц. Щербатий М. В.

Національна шкала\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів:\_\_\_\_\_\_\_\_\_\_ Оцінка ECTS:\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

Львів 2019

**Зміст**

Table of Contents

[Вступ 3](#_Toc10174642)

[1. Процес роботи з сурогатними моделями 4](#_Toc10174643)

[1.1 Вибір плану проведення експериментів 5](#_Toc10174644)

[1.1.1 Рівномірне розбиття простору 5](#_Toc10174645)

[1.1.2 Латинський гіперкуб 6](#_Toc10174646)

[2. Крігінг 11](#_Toc10174647)

[1.1 Побудова моделі Крігінгом 11](#_Toc10174648)

[2.2 Передбачення значень Крігінгом 14](#_Toc10174649)

[3. Програмна реалізація Крігінгу 16](#_Toc10174650)

[4. Результати виконання досліджень 17](#_Toc10174651)

[Висновки 19](#_Toc10174652)

[Список використаної літератури 20](#_Toc10174653)

# Вступ

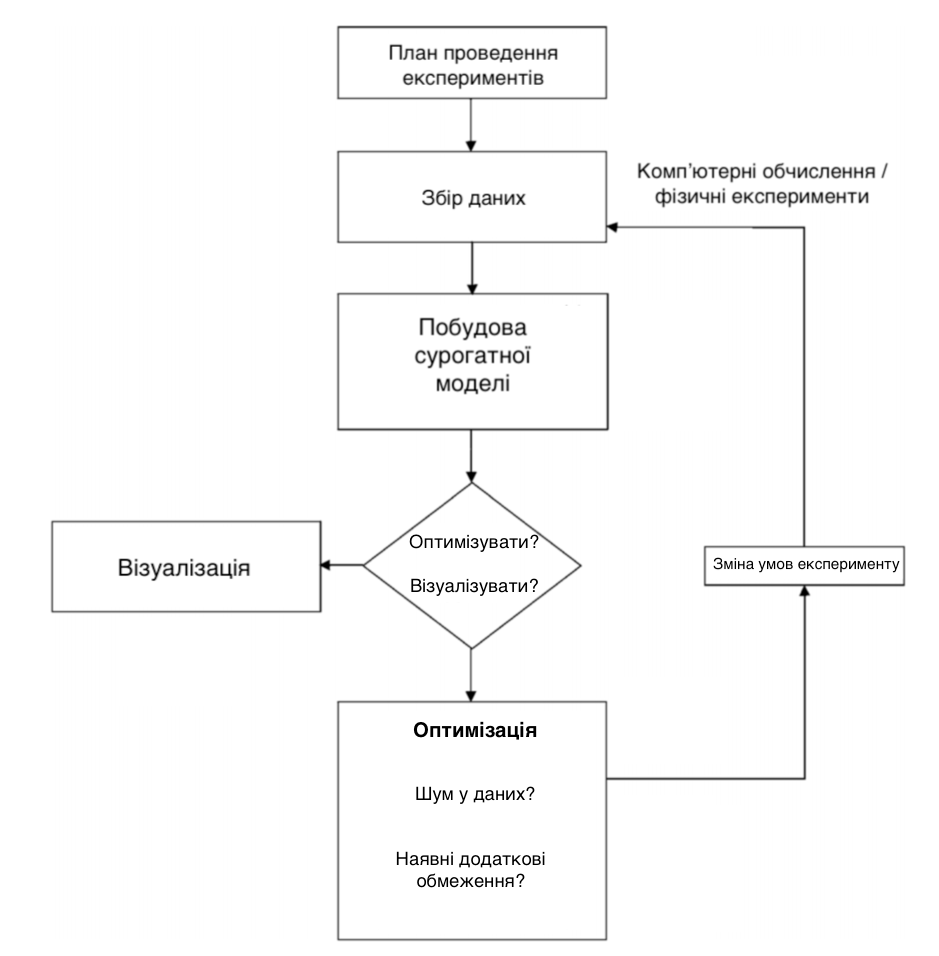
Одним із найпоширеніших застосувань математики у прикладних сферах – наприклад, інженерії – є використання математичних моделей для симуляції комплексних процесів. Водночас, основною проблемою залишається обчислювальна складність. Деякі обчислювальні експерименти можуть займати дуже великі об’єми часу (в рангах годин), і лінійний прогрес у сфері збільшення потужностей обчислювальних машин не в змозі задовільнити потребу у швидкому виконанні обчислень вищих порядків складності.

Проблема полягає також у тому, що під час процесу пошуку певних оптимальних значень для комплексної системи, характер дослідження змушує нас повторювати затратні обчислення багато разів на всій множині визначення вхідних параметрів, постійно шукаючи оптимальніше співвідношення між ними.

На таких міркуваннях базується ідея використання «сурогатної моделі» - знаходження швидкої для обчислення апроксимації для заданої складної математичної моделі. В цій роботі я розгляну основні етапи при роботі із сурогатними моделями, та опишу використання методу Крігінгу для побудови таких моделей.

# 1. Процес роботи з сурогатними моделями

Інженери використовують сурогатні моделі у багатьох сценаріях: прогноз поведінки складної для обчислень функції в області визначення; калібрація математичних моделей фізичних процесів з метою покращення точності без збільшення затрат на їх обчислення; очищення даних, отриманих внаслідок фізичних експериментів, від шуму; передбачення результатів для вхідних даних, для яких неможливо провести експеримент.

В усіх цих випадках алгоритм роботи з сурогатними моделями аналогічний. Його можна описати такою схемою: ****

## 1.1 Вибір плану проведення експериментів

Розглядатимемо певну неперервну функцію , де , для якої ми намагаємось побудувати сурогатну модель . Окрім припущення неперервності, все що ми можемо дізнатись про набір дискретних спостережень

.

Оскільки ці значення складні для обчислень, наше завдання – підібрати оптимальний для побудови хорошої моделі план проведення експериментів . Погано спроектований набір даних може привести до моделі, яка акуратна в своїх передбаченнях лише на певних ділянках .

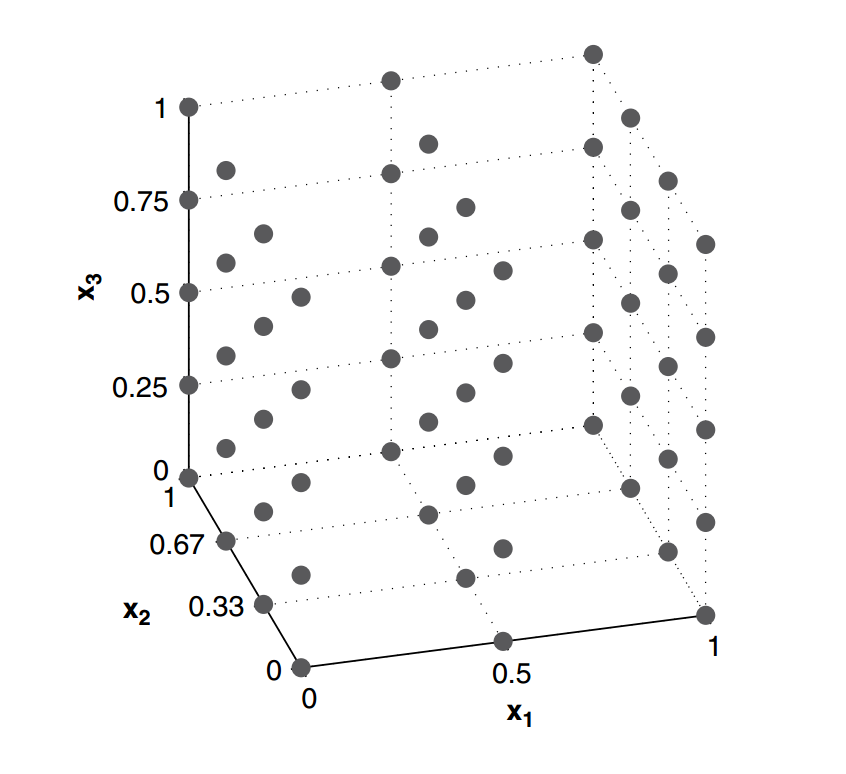
Загальною характеристикою всіх методів побудови сурогатних моделей є те, що вони значно точніші у своїх передбаченнях в околах Тому, основною метою при створенні плану проведення експериментів є рівномірний розподіл точок по всьому . План, що задовільняє такій вимозі, називатимемо таким, що заповнює простір (space-filling). (*Forrester A.I.J., Keane A.J. Engineering Design via Surrogate Modelling. 2008 – 1.4.1*)

### 1.1.1 Рівномірне розбиття простору

Найпростіший спосіб задати подібне розбиття простору – заповнити його -вимірною сіткою рівновіддалених точок. Такий план, очевидно, заповнить простір, однак він містить ряд недоліків.

По-перше, якщо для досягнення певної точності необхідно точок у одному вимірі, для простору кількість обчислень функції зростає до . Такий план називається повним факторним експериментом. Оскільки ми прагнемо мінімізувати затрати на обчислення при підготовці даних, така тенденція є великим мінусом.

По-друге, план, створений описаним методом, негнучкий у своєму заповненні простору. На діаграмі 1.1 зображено рівномірне розбиття трьохвимірного простору. При проекції на осі, точки з цього розбиття накладатимуться, тому можна стверджувати, що для кожної із трьох змінних план може бути покращений певним зсувом точок вдовж відповідної осі.

****

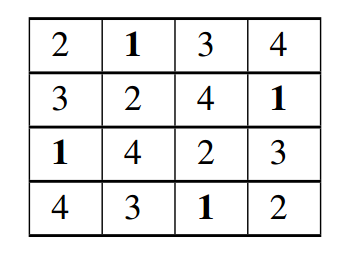
**Діаграма 1.1**

Цього можна досягти, розбивши відрізок допустимих значень цієї змінної на велику кількість рівних інтервалів, та заповнивши їх точками експерименту випадковим чином. Отриманий таким чином план називають *випадковою стратифікованою пробою*. Узагальнюючи цю ідею для всіх змінних, отримаємо техніку побудови плану експерименту, що називається *Латинський гіперкуб*.

### 1.1.2 Латинський гіперкуб

Для опису побудови Латинського гіперкубу для -вимірного випадку, спершу проілюструємо його побудову для двохвимірного простору.

Якщо для досягнення бажаної точності моделі необхідно точок проведення експерименту, простір розбивається на підпростори рівномірною сіткою, після чого кожен рядок заповняється перестановкою так, щоб кожне число з’являлось у кожному рядку і стовпці лише один раз (Діаграма 1.2).

**

**Діаграма 1.2**

Після цього з кожного підпростору, позначеного, наприклад, **1**, вибирається одна точка у план проведення експерименту довільним чином. Такий квадрат називається *Латинським квадратом*, його відповідний -вимірний аналог – Латинським гіперкубом.

Побудований таким методом план гарантує, що при проекції на жодну з осей точки не будуть накладатись одна на одну. Це, однак, не гарантує того, що кожен такий гіперкуб заповнюватиме простір: наприклад, ми можемо отримати куб, де всі точки експерименту розміщені на головній діагоналі. Щоб характеризувати те, наскільки гіперкуб заповнює простір, скористаємось метрикою *мінімакс* (*Johnson et al. 1990*).

Нехай унікальні значення відстаней між усіма можливими парами точок у , посортовані у порядку зростання; набір значень, де кількість пар точок з , відстань між якими рівна . Ми називатимемо ***мінімаксним*** із всіх можливих планів, якщо в нього максимальне можливе і, серед всіх планів для яких ця умова виконується, мінімальне .

Відстанню між двома точками у вважатимемо *p-норму*:

У літературі немає переконливих демонстрацій переваги норми для певного *p* у контексті оцінки мінімаксом (*Forrester A.I.J., Keane A.J.*). Оскільки обчислення даної норми є найлегші при , надалі використовуватимемо її, якщо не вказано інакше.

Варто зазначити, що таке означення не завжди однозначно визначає мінімаксний план. Тому, скористаємося доповненням до означення, запропонованих *Моррісом і Мітчелом у 1995*:

– ***мінімаксний*** із всіх можливих планів, якщо в нього максимальне можливе , серед всіх планів, для яких ця умова виконується, мінімальне , серед всіх планів, для яких ця умова виконується, максимальне , … , мінімальне

Залишається додати, що задля спрощення обчислень та уникнення проблем із великими числами, перед початком проведення будь-яких обчислень ми нормалізовуємо до одиничного кубу .

1. **2** Побудова сурогатної моделі

Отримавши з кроку **1.1** набір з пар значень , ми шукаємо функцію з простору функцій . Процес такого пошуку називається *контрольованим навчанням*.

Звісно, переважна більшість функцій з цього простору наближатимуть дуже погано; ми можемо значно звузити коло пошуку певними обмеженнями – наприклад, покласти що повинна бути неперервна – однак нам все одно доведеться обирати із безлічі функцій, що потенційно нам підходять.

Слід зазначити, що з метою тестування слід обрати певний набір тестових точок, незалежний від тренувального набору , та обчислити точне значення у цих точках. Його рекомендований обсяг - . При моделюванні реальних задач, ці додаткові затрати на обчислення потрібно враховувати при оцінці допустимого обсягу . Ці дані не слід використовувати під час тренування моделі, адже вони потрібні, щоб дати оцінку, наскільки добре передбачає значення у точках, на яких не проводилось навчання. Цей набір обиратимемо також за допомогою Латинського гіперкубу.

Для оцінки того, наскільки добре знайдена наближає , існує цілий ряд характеристик. Найчастіше вживаними із них є:

* Максимальне абсолютне відхилення ( розмір тестового набору даних):
* Середньоквадратичне відхилення:
* Коефіцієнт кореляції:

Щодо RMSE, можна вважати, що якщо його значення, поділене на розкид значень серед тестових , менше *0,1* (ця величина матиме значення від 0 до 1), модель дає достатньо хороші передбачення. Метрика не залежить від області значень *f* ; вона набуває значень від 0 до 1 та характеризує топологію функцій, не їх конкретні значення. Чим ближче до 1, тим краще модель передбачує функцію; приблизно відповідає RMSE = 0.1.

# Крігінг

Метод *Крігінгу* – частковий випадок методу *радіальних базисних функцій* з базисною функцією виду

(Опис методу радіальних базисних функцій (надалі РБФ) знаходиться за межами даної роботи, його можна знайти у, наприклад, *Zhong-Hua Han and Ke-Shi Zhang (2012). Surrogate-Based Optimization, Real-World Applications of Genetic Algorithms*).

Особлива відмінність цієї базисної функції – наявність наборів коефіцієнтів та , що дозволяє їй змінювати свою поведінку для кожної змінної. Наприклад, при та константою для всіх просторів, метод Крігінга перетворюється у метод РБФ з базисною функцією Гауса. Таким чином, Крігінг накладає найменше припущень на завдяки більшій кількості параметрів моделі – припускається, що вона гладка та нерозривна.

Варто зазначити, що розглядається *інтерполяція Крігінгом.* Відома в топології *регресія Крігінгом* лежить поза межами даної роботи.

## 1.1 Побудова моделі Крігінгом

Ідея методу Крігінгу полягає в тому, що ми розглядаємо отримані значення як значення, отримані внаслідок стохастичного (випадкового) процесу. Позначимо це набором випадкових векторів

Нехай цей набір має середнє значення , вектор-стовпець одиниць. Випадкові змінні корелюються одна з одною через базисну функцію:

Відповідно, ми можемо побудувати матрицю кореляції для всіх вхідних даних:

Саме для введення такої кореляції між даними нам необхідне припущення про гладкість та неперервність *f*. Значення у матриці залежать від абсолютних відстаней між точками та параметрів і . Більші значення відповідають більшій кореляції між точками по *j*-ій змінній; параметри характеризують вплив *j*-ої змінної на Y чим більший , тим більший вплив має на значення Y *j*-та змінна.

Щоб побудувати сурогатну модель, ми намагатимемося обрати такі і , щоб максимізувати вірогідність **y**, отримуючи таким чином , що максимізує вірогідність отримати правильні значення *y* на всьому .

Для подальших міркувань, опишемо процес оцінки моделі за допомогою методу максимальної вірогідності.

Вірогідність – величина, що характеризує імовірність того, що набір даних було отримано із функції , де певний невеликий сталий окіл допустимої похибки кожного із значень. Наприклад, припустивши що випадково незалежно розподілені згідно нормального розподілу із стандартним відхиленням , вірогідність набору даних є

Ми прагнемо максимізувати правдоподібність моделі для набору даних **x**, а для зручності обчислень – мінімізувати її від’ємний натуральний логарифм:

Оскільки ми припускаємо що модель, побудована інтерполяцією Крігінга, не містить похибки , імовірність можна записати наступним чином:

Що можна перевиразити через наші дані як

Запишемо логарифм, який необхідно максимізувати:

Прирівнюючи похідні рівняння (2.3) до нуля, отримуємо наступні найбільші вірогідності для і

Тепер ми можемо підставити отримані значення назад у (2.3) і прибрати доданки-константи для отримання *функції концентрованої логарифмічної вірогідності*:

Значення цієї функції залежить від шуканих нами і Для її максимізації, оптимально буде використати техніку числової оптимізації – наприклад, глобальний пошук, що уже реалізований в середовищі Matlab. (*Forrester A.I.J., Keane A.J.*)

## 2.2 Передбачення значень функції Крігінгом

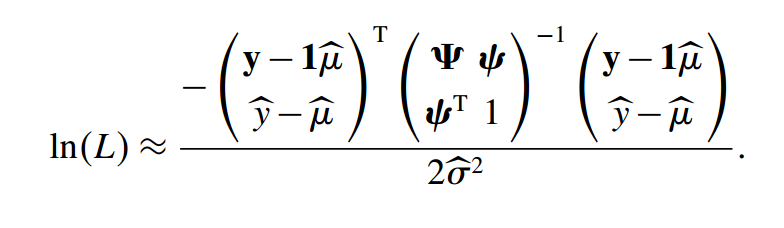
Кожне нове значення , передбачене моделлю Крігінгу, повинне сходитися з уже обчисленими для тренувального набору даних параметрами кореляції. Тому, ми додаємо до нового вектора та означаємо новий вектор кореляцій між тренувальними даними та

Тепер, сконструюємо доповнену матрицю кореляцій:

Останній елемент одиниця, адже головна діагональ заповнена одиницями ().

Запишемо логарифмічну імовірність доповнених даних:

Тільки останній доданок цього виразу залежить від тому ми можемо розглядати тільки його для максимізації. Підставивши значення і отримаємо

****

Щоб максимізувати цю рівність, нам треба обчислити Для цього, можна використати, наприклад, метод знаходження оберненої матриці Тейла (*1971*). Знайшовши розклад через та виконавши відповідні спрощення, знаходимо, що найкраща вірогідність для досягається при

Не одразу є очевидним, як рівняння (2.4) випливає з РБФ (2.1). У рівнянні (2.4), базисні функції містяться у векторі Модель побудована таким чином, що вона проходить через усі точки вхідних даних (інтерполює їх) : якщо ми спробуємо передбачити значення для ми отримаємо значення

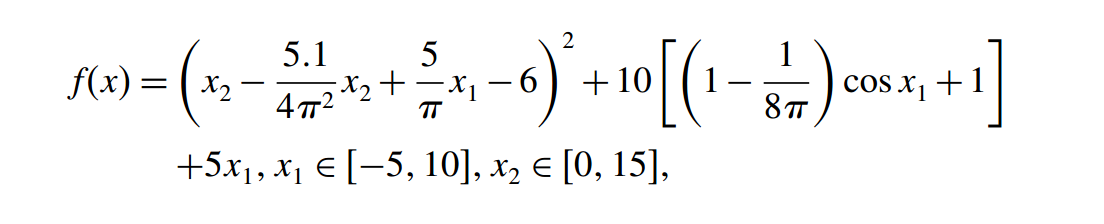
# 3. Програмна реалізація Крігінгу

Програмна реалізація методу Крігінгу виконана у середовищі Matlab. Для роботи з програмою використовується консольний користувацький інтерфейс (CUI).  
Для зручності користування та підтримання чистоти коду рішення, вся інформація про модель, що будується, зберігається у глобальній структурі **ModelInfo**. Процес роботи з програмою повторює описаний у розділі **1** процес роботи з сурогатними моделями:

1. На цьому етапі нам необхідно занести інформацію про розмірність *D* та бажану кількість точок у тренувальному та тестувальному наборах. Також, необхідно вказати на функцію, яку ми наближатимемо моделлю, для отримання точних результатів. Після цього, запустивши скрипт **ModelInfo.m**, програма згенерує оптимальний Латинський гіперкуб для обох наборів, та обчислить значення функції у відповідних точках.
2. Побудова моделі – маючи дані з пункту 1, побудуємо матрицю та збережемо її у **ModelInfo**; також збережемо обраховані значення . Для спрощення програмної реалізації, ми не розглядатимемо вектор ***p*** як змінний параметр, а приймемо усі
3. Маючи усі необхідні дані занесеними у **ModelInfo**, ми можемо протестувати нашу модель. Передбачення результату у точці виконуватимемо за допомогою функції **pred(x)**: вона використовуватиме тренувальний набір даних, а також матрицю та вектор для обрахунку за описаною вище формулою.

# 4. Результати виконання досліджень

1. Проведемо експеримент для функції двох змінних, намагаючись натренувати сурогатну модель для ф-ії Браніна



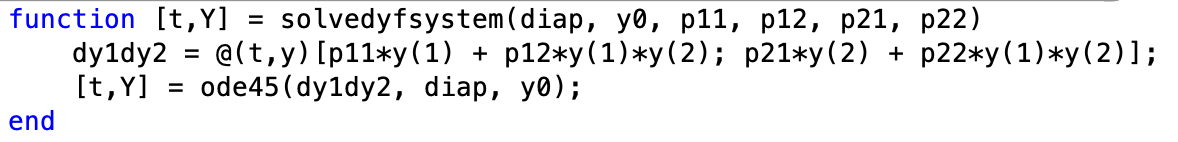
Для оцінки якості моделі, рахуватимемо максимальну абсолютну похибку та нормалізовану середню квадратичну похибку на тестовому наборі даних. Зберемо ці дані для *nTrain* = 10, 20, 50, 80 і 100, та порівняємо результати:

|  |  |  |  |
| --- | --- | --- | --- |
| nTrain | nTest | MAD | RMSE |
| 10 | 2 | 2.532411e+01 | 1.944342e-01 |
| 20 | 4 | 2.773542e+00 | 3.760100e-02 |
| 50 | 12 | 1.734680e-03 | 3.527605e-06 |
| 80 | 20 | 5.474167e-04 | 9.840761e-07 |
| 100 | 25 | 7.502462e-04 | 1.919211e-06 |

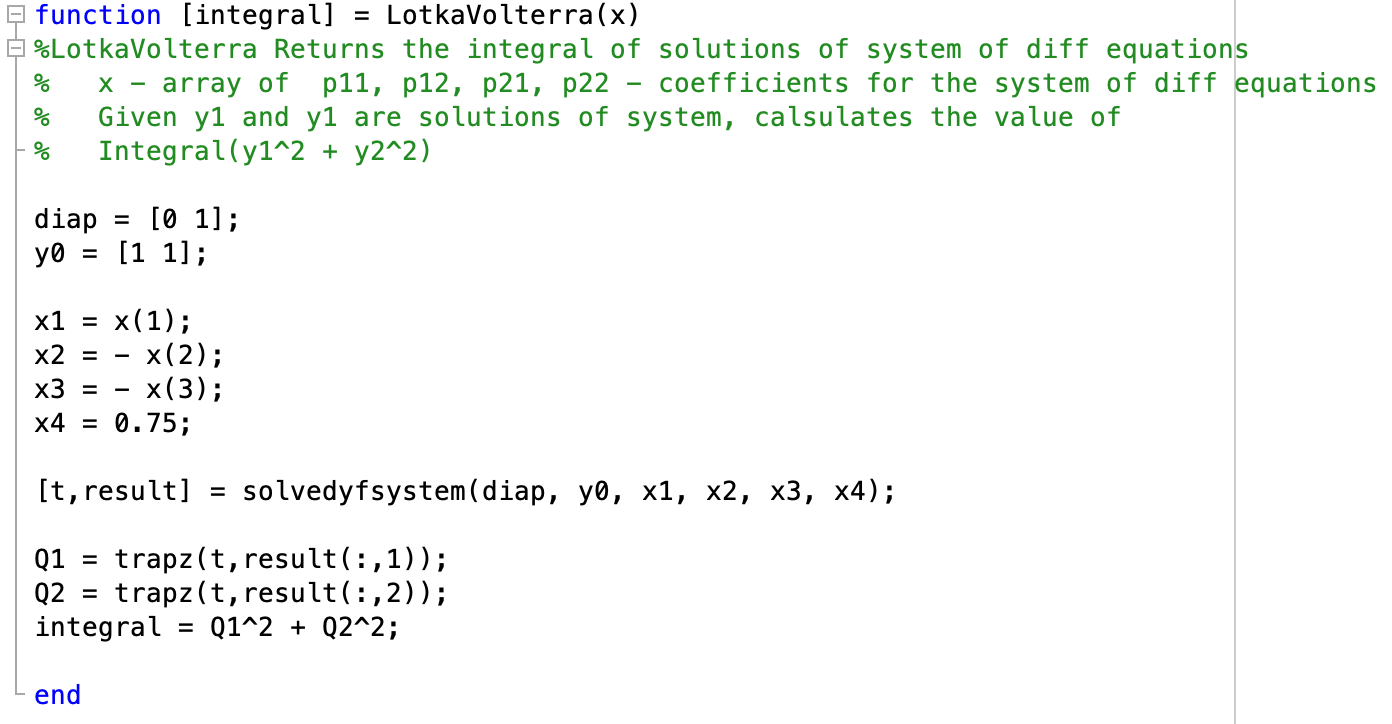
Можна бачити, що через випадкову природу вибору тренувального набору, після досягнення певної точності збільшення кількості даних не обов’язково призводитиме до збільшення точності моделі, тобто зростання точності при збільшенні тренувального набору досить незначне і може бути скасованим недетермінованістю вибору. RMSE порядку однак, уже можна вважати дуже хорошою точністю наближення.

2. Проведемо експеримент, використовуючи модель Лотки-Вольтерри:

Для розв’язування системи диференційних рівнянь використовуватимемо функцію solvedyfsystem, що прийматиме діапазон [], вектор початкових значень y0, та 4 значення параметрів системи, і розв’язує систему за допомогою функції Matlab ode45.



Для експерименту, створимо функцію LotkaVolterra, в якій задамо усі параметри системи: діапазон *t* = [0,1], початкові значення Оскільки всі коефіцієнти *p* належать проміжкові [0,1], ми домножимо коефіцієнти і на .  
Для того, щоб отримувати скалярний результат із розв’язків системи , ми рахуватимемо значення



Проведемо експеримент, аналогічний попередньому:

|  |  |  |  |
| --- | --- | --- | --- |
| nTrain | nTest | MAD | RMSE |
| 10 | 2 | 1.222480e-01 | 1.268602e-01 |
| 20 | 4 | 4.124557e-02 | 9.701197e-03 |
| 50 | 12 | 1.105292e-02 | 1.940622e-03 |
| 80 | 20 | 3.443630e-03 | 6.315950e-04 |
| 100 | 25 | 1.616825e-03 | 2.359096e-04 |

# Висновки

Метод інтерполяції Крігінга – один із найуніверсальніших методів побудови сурогатних моделей. Будучи одним із варіантів реалізації методу радіальних базисних функцій, він надає краще узагальнення апроксимованої функції завдяки додатковим наборам параметрів.

Для протестованих мною функцій, побудована за допомогою Крігінгу сурогатна модель показувала дуже хороші результати апроксимації (RMSE < 0.1) уже при 20 точках у навчальному наборі даних; при збільшенні цього набору, точність моделі зростала відповідно. Через випадкову природу формування тренувального набору даних, однак, після досягнення певної точності збільшення кількості точок не гарантуватиме збільшення точності апроксимації; потрібні значні збільшення *nTest* для значимого збільшення точності моделі.

# Список використаної літератури

1. *Forrester A.I.J., Keane A.J. Engineering Design via Surrogate Modelling. 2008*
2. *Zhong-Hua Han and Ke-Shi Zhang (2012). Surrogate-Based Optimization, Real-World Applications of Genetic Algorithms*
3. *Shawn E. Gano, Harold Kim and Don E. Brown. Comparison of Three Surrogate Modeling Techniques: Datascape, Kriging, and Second Order Regression. 2006*
4. *Yew S. Ong, Prasanth B. Nair and Andrew J. Keane. Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling. 2003*
5. *Alberto Serna , Christian Bucher. Advanced Surrogate Models for*

*Multidisciplinary Design Optimization. 2009*