

Синтезовано центральний композиційний ротатбельний план другого порядку та отримано адекватну математичну модель процесу наповнення-пластифікації шкіряного матеріалу. Модифікований метод релаксації використаний для відшукування оптимальних значень цільової функції даного процесу.

### Література

1. *Інноваційні технології виробництва шкіряних і хутрових матеріалів та виробів*: монографія / за ред. А. Г. Данилковича. – К. : Фенікс, 2012. - 344 с.
2. *Грищенко, І. М. Поліфункціональні шкіряні матеріали*: монографія / І. М. Грищенко, А. Г. Данилкович, О. Р. Мокроусова ; за ред. А. Г. Данилковича. – К. : Фенікс, 2013. – 268 с.

### АПРОКСИМАЦІЯ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ У JUPITER

Біла К. О., Концевой С. А.

### АППРОКСИМАЦИЯ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ В JUPITER

Белая Е. А., Концевой С. А.

### APPROXIMATION OF EXPERIMENTAL DATA IN JUPITER

Bila K., Kontsevoi S.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Київ, Україна  
[katebila7@gmail.com](mailto:katebila7@gmail.com)

*Представлено апроксимацію експериментальних даних, яка реалізована у середовищі Jupiter на мові Python з використанням бібліотеки numpy. Основна відмінність запропонованого підходу від існуючих (Excel та Qbasic) полягає у програмному обранні функції, яка забезпечує найкращу точність апроксимації на основі відносної похибки. Вказано можливість застосування цього підходу у «хмарному» середовищі Azure Microsoft, яке, як і локальне (пакет Anaconda), є безкоштовним.*

**Ключові слова:** Експериментальні дані, апроксимація, програмований вибір, Python, Jupiter, Numpy

*Представлено аппроксимацию экспериментальных данных, реализованная в среде Jupiter на языке Python с использованием библиотеки numpy. Основное отличие предлагаемого подхода от существующих (Excel и Qbasic) заключается в программном выборе функции, которая обеспечивает лучшую точность аппроксимации на основе относительной погрешности. Указано возможность применения этого подхода в «облачной» среде Azure Microsoft, которая, как и локальная (пакет Anaconda), является бесплатной.*

**Ключевые слова:** Экспериментальные данные, аппроксимация, программируемый выбор, Python, Jupiter, Numpy

*An approximation of experimental data, which is implemented in the Jupiter environment with Python's library numpy, is presented. The main difference between the proposed approach and the existing (Excel and Qbasic) is the programmed selection of function, which provides the best approximation accuracy based on relative error. The possibility of using this approach in the "cloud" environment (Azure Microsoft) is discussed, which as the local (based on Anaconda package) is free.*

**Keywords:** Experimental data, approximation, programmable selection, Python, Jupiter, Numpy

Математична обробка даних досліджень хімічних процесів (зазвичай залежність концентрацій або ступеня перетворення від часу) є необхідною складовою аналізу експериментальних результатів. Для економії часу і збільшення точності виникає необхідність в використанні готових рішень таких задач. Для цього на кафедрі ТНР, В та ЗХТ хіміко-технологічного факультету було створено програму у середовищі *Qbasic* (у до ексельну епоху), при цьому як і у середовищі *Excel* обрання відповідної функції за результатами апроксимації (близько 20-ма функціями) виконує сам дослідник.

Завданням цієї роботи було автоматизувати вибір функції, яка найкраще відповідає вихідним даним. Задачу апроксимації реалізовано в середовищі *Jupyter notebook* (наступник *iPython*) з інсталяційного пакету *Anaconda* [1], що зручно для використання завдяки безкоштовній доступності цього пакету. Також було використано відповідні бібліотеки *Python*. Для апроксимації ступеневої експоненціальної функції було використано числовий алгоритм з бібліотеки *numpy* – *curve\_fit()*. Для апроксимації полінома використано алгоритм – *np.polyfit()*. Також реалізовано алгоритм по методу найменших квадратів, але в кінцевій програмі використано саме функція *np.polyfit()*, щоб зменшити число строк в програмі. Було знайдено похибки апроксимації і значення середньої похибки для кожної функції за допомогою *np.linalg.norm()*.

За значення найменшої середньої похибки апроксимації у програмі обирається функція, яка видається користувачу програми. Також програма представляє графік отриманої функції і вихідної залежності, так що можливо оцінити похибку візуально.

Фрагмент програми (у середовищі *Jupiter*) апроксимації ступеневої функції представлено нижче, а вихідні дані та результат її роботи - на рисунку.

Таким чином, створена програма пришвидшує вибір апроксимаційної функції результатів експерименту на основі порівняльного аналізу відносної похибки кожної можливої функції. При цьому, усі необхідні програмні компоненти цього рішення є легально безкоштовними.

На цей час середовище *Jupiter* з підтримкою *Python*, *R* та *F#* є доступним безкоштовно у "хмарному" середовищі Microsoft [2], що дозволяє виконувати ту ж саму програму у форматі *IPython Notebook (.ipynb)* без інсталяції необхідного програмного забезпечення.

```
In [6]: def func_powerlaw(x, m, c, c0):  
        return c0 + x**m * c  
  
popt, pcov = curve_fit(func_powerlaw, x, y, p0 = np.asarray([-1,10**5,0])  
))  
print popt  
  
errors = y - func_powerlaw(x,*popt)  
print "Похибки в точках: ", errors  
error = np.linalg.norm(errors)  
print "Загальна похибка: ", error  
  
M = popt[0]  
C = popt[1]  
C0 = popt[2]  
print "Отримане рівняння ----solution:"  
print "%f x ^ %f + %f = y" % (C, M, C0)  
Out[6]: [<matplotlib.lines.Line2D at 0x7f003ec31510>]
```

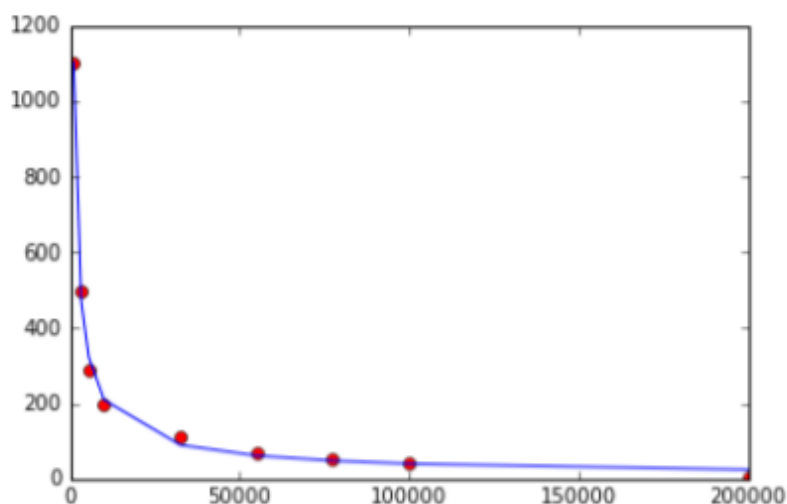


Рис. Графік обраної функції і вихідних даних

## Література

1. *Anaconda*. The Most Popular Python Data Science Platform. Режим доступу: <https://www.anaconda.com/download/>.
2. *Microsoft Azure Notebooks*. Режим доступу: <https://notebooks.azure.com/>.