

Розробка лекційних демонстрацій засобами MMC Sage

Постановка проблеми. Модернізація освіти неможлива без впровадження в учбово-виховний процес інформаційно-комунікаційних технологій (ІКТ). Загальними засобами інформатизації освіти є апаратне забезпечення, програмне забезпечення та змістовне наповнення [1].

Ефективність використання інформаційних технологій залежить від способів та форм їх застосування, надаючи можливість:

- індивідуалізувати навчальний процес зі збереженням його цілісності та контрольованості;
- покращити сприйняття студентами навчальних відомостей та активізувати їхню пізнавальну діяльність.

До ІКТН відносяться системи комп'ютерної математики (СКМ), що традиційно використовуються як засіб для підтримки практичних та лабораторних занять.

Сучасні СКМ – мобільні математичні середовища (MMC), зокрема Sage, надають можливість створення інтерфейсу користувача, що дозволяє розглядати їх як засіб для створення лекційних демонстрацій.

Призначення статті: розглянути особливості розробки лекційних демонстрацій в середовищі MMC Sage.

Основна частина. Для створення демонстраційних програм з напівавтоматичним або автоматичним режимом роботи, в яких передбачається багаторазове виконання обчислень для різних значень вхідних параметрів, використовують візуальні елементи управління типу «поле для введення», «ковзунок», «прапорець» та інші.

Для визначення і додавання кожного елемента управління в Sage призначена відповідна функція. Виклик такої функції здійснюється в полі опису параметрів деякої функції користувача, якій передує вказівка `@interact`, наприклад:

```
@interact
def name_user_func1(f=input_box\
(x^5-x^2+1,label="Функція:")):
    print show(f)
```

Розглянемо функції для створення найбільш поширених елементів управління [2].

Для визначення й додавання елемента управління типу «поле для введення» в Sage призначена функція `input_box(default, label, type)`

де `default` – значення, що повертається за замовчуванням;

`label` – напис ліворуч від елемента;

`type` – поле для введення вхідних даних у вигляді довільного рядка.

Для створення елемента управління типу «прапорець» в SAGE призначена функція `checkbox(default, label)`,

де `default` може мати значення `true` або `false`,

`label` – назва «прапорця».

Для створення елемента «прапорець» у скороченому форматі достатньо вказати стан перемикача (увімкнений/вимкнений) – `cb1=true` або `cb2=false`.

Для створення елемента управління типу «меню вибору» в SAGE призначена функція `selector(values, [label], [nrows], [ncols], [buttons])`:

де `values` – значення пунктів меню вибору, що можуть зазначитися переліком елементів – `[val1, val2, val3, ...]` або діапазоном елементів – `[val_start..val_finish]`;

`label` – надпис ліворуч від елемента;

`nrows` – кількість рядків у поданні пунктів меню вибору (при поданні пунктів меню вибору у вигляді кнопок);

`ncols` – кількість колонок у поданні пунктів меню вибору (при поданні пунктів меню вибору у вигляді кнопок);

`buttons` – додатковий параметр логічного типу: при встановленому значенні `true` меню вибору подається у вигляді кнопок, при встановленому значенні `false` (за замовчуванням) – у вигляді випадаючого списку.

Розглянемо застосування MMC Sage на прикладі лекції «Моделювання випадкових величин та випадкових подій». В даній лекції, зокрема розглядаються різні методи генерації псевдовипадкових величин, розподілених рівномірно (методи середини квадрату, лінійний конгруентний, вбудований), нормально (метод полярних координат) та експоненційно (метод логарифму).

Для створення лекційної демонстрації застосуємо модуль `finance`, який було розроблено Вільямом Штейном [3] для роботи з фінансовими часовими рядами.

Для демонстрації випадкових подій доцільно використовувати модулі `TimeSeries` і `Stock`.

Модуль `TimeSeries` призначено для роботи з дискретними часовими рядами з плаваючою крапкою.

Модуль `Stock` – клас для отримання даних з фондової біржі в реальному часі. В даному класі реалізовані функції для завантаження часових рядів з мережі Інтернет або з деякого файлового ресурсу.

Розглянемо основні методи, які використовуються для отримання характеристик часових рядів:

`diffs()` – функція повертає новий часовий ряд, отриманий за допомогою знаходження різниці між поточним елементом та попереднім. Таким чином, якщо існує наступний часовий ряд X_0, X_1, X_2, \dots , то за допомогою даної функції отримаємо новий часовий ряд $X_1 - X_0, X_2 - X_1, \dots$. Вихідний часовий ряд має на один елемент менше, ніж початковий ряд.

`histogram()` – повертає частотну гістограму. Кожний інтервал гістограми являє собою кількість попадань значень даної серії в інтервал. Кількість інтервалів визначається вхідним параметром `bins` – додатним числом (за замовчуванням 50).

`plot_histogram()` – повертає зображення гістограми для даного часового ряду із заданою кількістю інтервалів.

`sums()` – повертає новий часовий ряд, отриманий за допомогою додавання всіх попередніх елементів.

`add_scalar()` – повертає новий часовий ряд шляхом додавання скаляра до кожного елемента ряду.

`standart_deviation()` – повертає стандартне відхилення вхідної послідовності.

`mean()` – повертає середнє значення елементів послідовності.

`randomize()` – генерує випадковий часовий ряд.

`plot()` – функція повертає зображення часового ряду у вигляді лінії або точок $(i, T(i))$, де i – невід’ємний діапазон цілих чисел від 0 до довжини часового ряду.

`autocorrelation()` – повертає кореляцію часового ряду самого з собою (між попередніми та наступними значеннями). Автокореляцію іноді називають *послідовною кореляцією*, що означає кореляцію між членами ряду чисел, розташованих у певному порядку. Також синонімами цього терміну є *лагова кореляція* та *персистентність*.

`central_moment()` – повертає центральний момент k -го порядку випадкової величини ξ , який є математичним сподіванням величини $(\xi - M\xi)^k$:

$$\mu_k = \mu_k(\xi) = M(\xi - M\xi)^k$$

`moment()` – повертає початковий момент k -го порядку випадкової величини ξ , який є математичним сподіванням величини ξ^k

$$\alpha_k = \alpha_k(\xi) = M(\xi^k)$$

`max()` – повертає максимальне значення часового ряду.

`min()` – повертає мінімальне значення часового ряду.

`range_statistic()` – повертає інтервал, який представляє собою різницю між максимальним та мінімальним значенням цього ряду, поділену на стандартне відхилення різниці між цими елементами.

Розглянемо структуру лекційної демонстрації (рис. 1):

```
def msk(x): #метод середини квадрату
#нове псевдовипадкове число в діапазоні [0;10000)
    return int(x^2/100.0)%10000
```

```
def lkm(x,a,c,m): #лінійний конгруентний метод
#нове псевдовипадкове число в діапазоні [0;m)
return (a*x+c)%m
```

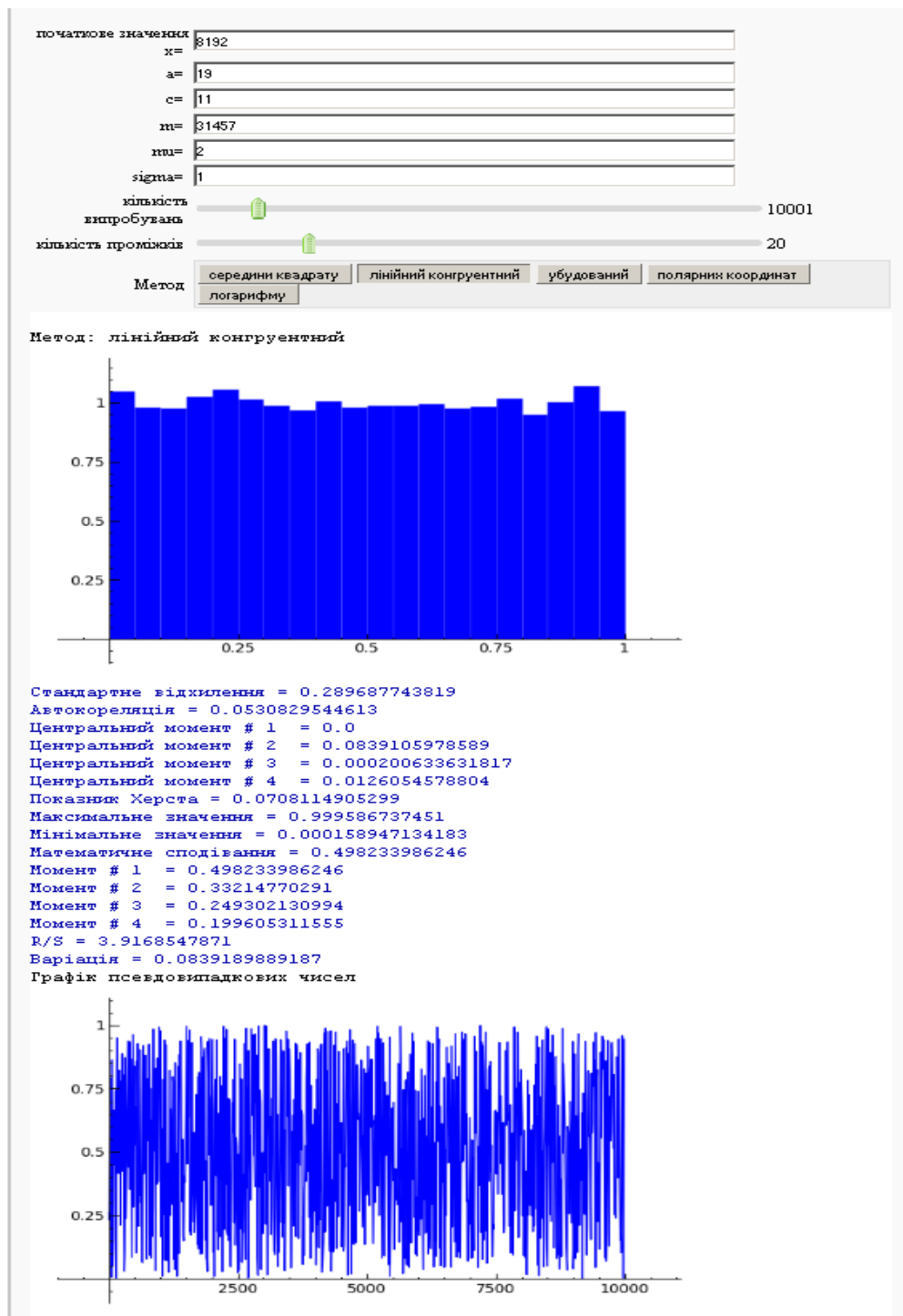


Рис. 1. Інтерфейс лекційної демонстрації

```
def mpk(mu,sigma): #метод полярних координат
s=1
while s>=1:
#два рівномірні розподіли ймовірностей на проміжку [0;1)
u1=random()
u2=random()
```

```
#два рівномірні розподіли ймовірностей на проміжку [-1;1)
```

```
    v1=2*u1-1          v2=2*u2-1
```

```
    s=v1^2+v2^2
```

```
    x1=v1*sqrt(-2*log(s)/s)
```

```
    x2=v2*sqrt(-2*log(s)/s)
```

```
    y1=mu+sigma*x1
```

```
    y2=mu+sigma*x2
```

```
    return y1
```

```
def ml(mu): #метод логарифму (показниковий розподіл)
```

```
    u=random()
```

```
    return -mu*log(u)
```

```
@interact
```

```
def _(x=input_box(default=8192,\
```

```
label="початкове значення x="),\
```

```
a=input_box(default=19, label="a="),\
```

```
c=input_box(default=11, label="c="),\
```

```
m=input_box(default=31457, label="m="),\
```

```
mu=input_box(default=2, label="mu="),\
```

```
sigma=input_box(default=1, label="sigma="),\
```

```
    n=slider(vmin=1, vmax=100000,step_size=100,\
```

```
default=10000,label="кількість випробувань"),\
```

```
    kol=slider(vmin=1, vmax=100,step_size=1,\
```

```
default=20,label="кількість проміжків"),\
```

```
    b = selector(["середини квадрату",\
```

```
                "лінійний конгруентний",\
```

```
                "вбудований",\
```

```
                "полярних координат",\
```

```
                "логарифму"], \
```

```
    label="Метод", buttons=True)):
```

```
    #список для накопичення згенерованих значень
```

```
    arr0=[]
```

```
    html("Метод: "+b)
```

```

for i in range(0,n):
    if b=="середини квадрату":
        #псевдовипадкове число в діапазоні [0;1)
        y=x/10000.0
        x=msk(x)
    elif b=="лінійний конгруентний":
        #псевдовипадкове число в діапазоні [0;1)
        y=x/m
        #нове псевдовипадкове число в діапазоні\
        #[0;m-1]
        x=lkm(x, a, c, m)
    elif b=="вбудований":
        y=random()
        #псевдовипадкове число в діапазоні [0;1)
    elif b=="полярних координат":
        y=mpk(mu, sigma)
    elif b=="логарифму":
        y=ml(mu)
        #додавання до списку згенерованого значення
        arr0.append(y)

series=finance.TimeSeries(arr0)
series.plot_histogram(kol).show()
print "Стандартне відхилення =", \
        series.standard_deviation()
print "Автокореляція =", series.autocorrelation()
for i in range(1,5):
    print "Центральний момент #", \
            i, " =", series.central_moment(i)
print "Показник Херста =", series.hurst_exponent()
print "Максимальне значення =", series.max()
print "Мінімальне значення =", series.min()
print "Математичне сподівання =", series.mean()
for i in range(1,5):
    print "Момент #", i, " =", series.moment(i)

```

```

print "R/S =",series.range_statistic()

print "Варіація =",series.variance()

html("Графік псевдовипадкових чисел")

series.plot().show()

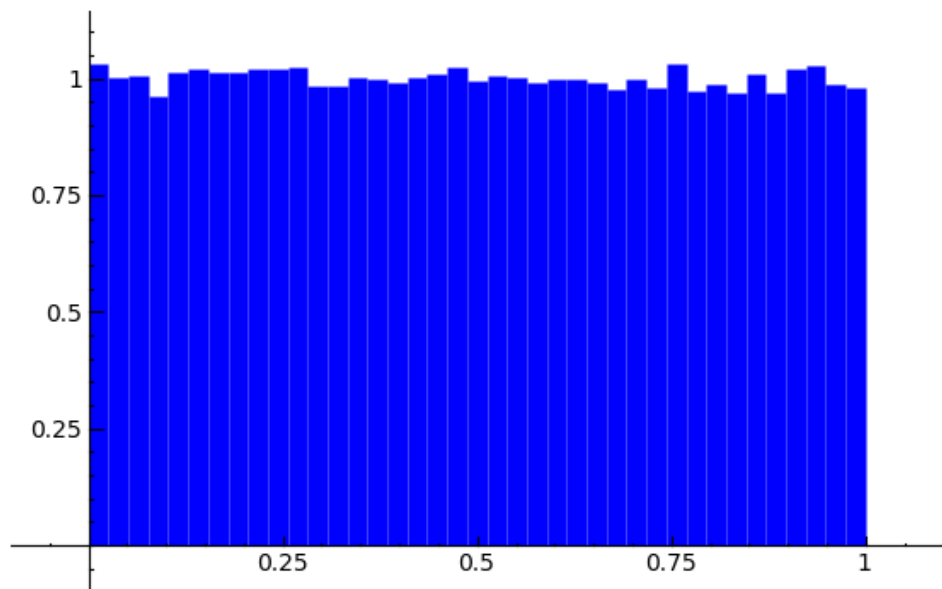
```

Початкова форма містить «поле для введення» та «ковзунок» для отримання початкових даних та кнопки вибору методу генерації псевдовипадкових чисел (рис. 2).

Після введення даних та вибору методу генерації отримаємо частотну гістограму (рис. 3а) та графік псевдовипадкових чисел (рис. 3б).

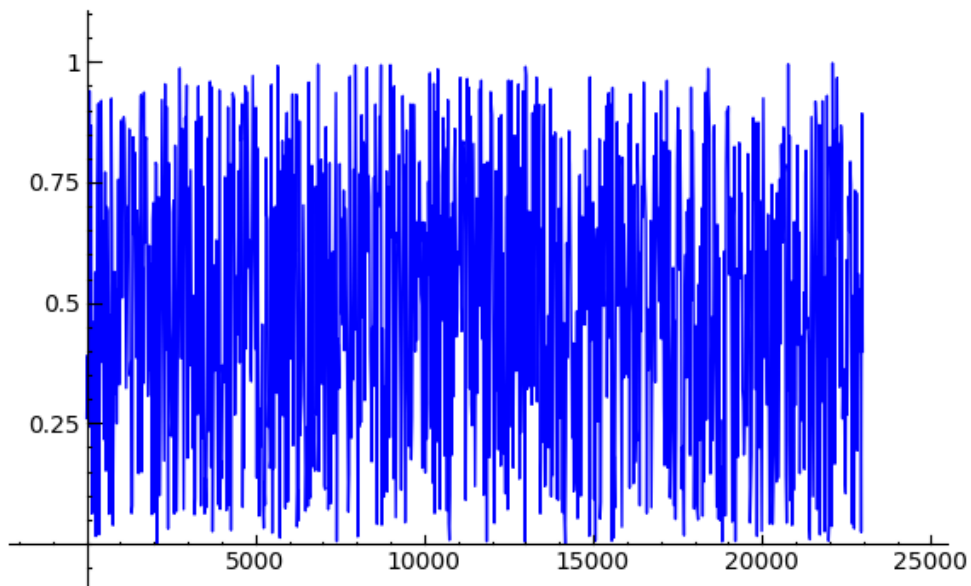
Рис. 2. Початкова форма введення даних

Метод: лінійний конгруентний



а) частотна гістограма

Графік псевдовипадкових чисел



б) графік псевдовипадкових чисел

Рис. 3. Ілюстрація лінійного конгруентного методу

```
Стандартне відхилення = 0.288779782022
Автокореляція = 0.0513244266884
Центральний момент # 1 = 0.0
Центральний момент # 2 = 0.0833901368464
Центральний момент # 3 = 0.00019264219582
Центральний момент # 4 = 0.0125138564739
Показник Херста = 0.0601402861861
Максимальне значення = 0.999968210573
Мінімальне значення = 0.000158947134183
Математичне сподівання = 0.498460306663
Момент # 1 = 0.498460306663
Момент # 2 = 0.331852814165
Момент # 3 = 0.248741444082
Момент # 4 = 0.198947676423
R/S = 3.92085665927
Варіація = 0.0833937625045
```

Рис. 4. Деякі числові характеристики часового ряду,
отриманого за допомогою лінійного конгруентного методу

Розглянута демонстрація дозволяє в ході лекції не лише продемонструвати характерні властивості послідовностей псевдовипадкових чисел, згенерованих за різними методами, а й виконати повноцінне їх дослідження в процесі обчислювального експерименту.

Література

1. Триус Ю. В. Нові інформаційні технології у навчальному процесі вищої школи / Ю. В. Триус // Сборник трудов четвертого научно-методического семинара «Информационные технологии в учебном процессе». – Одесса: ЮГПУ им. К. Д. Ушинского, 2003. – С. 159–161.
2. Шокалюк С. В. Основи роботи в SAGE / С. В. Шокалюк; за ред. академіка АПН України М. І. Жалдака. – К.: НПУ імені М. П. Драгоманова, 2008. – 64 с.
3. Stein W. Sage Programming Guide. / Stein W., Joyner D. – 2008. – 86 p.