

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ СИСТЕМ І ПРОЦЕСІВ

Лекція 1. Вступ до сучасного математичного моделювання

Київ – 2015

Задача цього курсу поглиблення знань щодо математичного моделювання систем та процесів різної природи, та реалізація їх в сучасними програмними засобами, без посередньо системами автоматизації програмування такими як МАХІМА та MathCAD. Більшість навчального матеріалу присвячена використанню сучасних підходів до побудови, аналізу так званих фрактальних структур. На лабораторних заняттях буде запропоновано реалізація теоретичних підходів у вигляді програмних реалізацій.

Базовою літературою для нашого курсу буде книга “Фракталы и хаос в динамических системах. Основы теории”[1].

Кроновер, Р. М. Фракталы и хаос в динамических системах. Основы теории / Р. М. Кроновер. — М. : Постмаркет, 2000. — 352 с.

Для початку розглянемо приклад. Візьмемо відому функцію
Вєрштра

$$W(x) = \sum_{n=0}^{\infty} w^n \cos(2\pi b^n x),$$

яка за своїми властивостями є скрізь
безперервною й недиференційованою.

В системі МАХІМА рядок для відображення її графіка буде
виглядати:

```
plot2d(sum(0.5^n*cos(2*\%pi*4^n*x),n,0,200),  
[x, 0, 0.5])$
```

В результаті отримуємо графік, який зображений на малюнку 1.1.

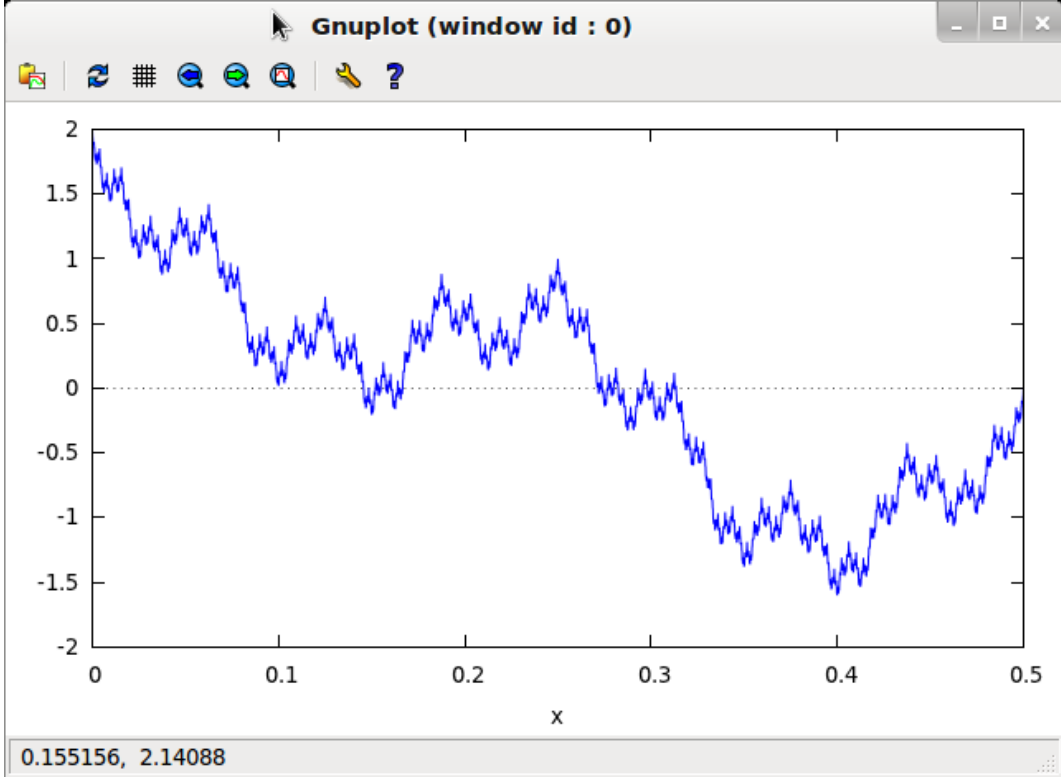


Рис.1.1. Функція Веєрштраса.

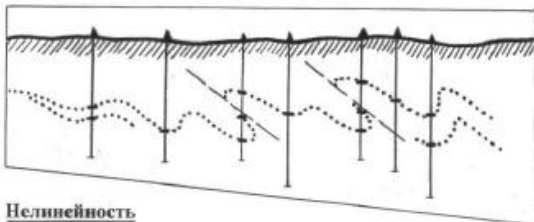
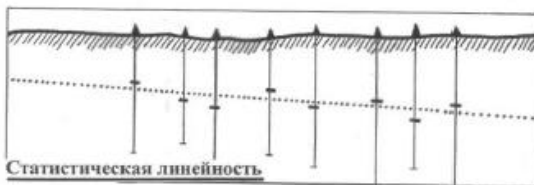
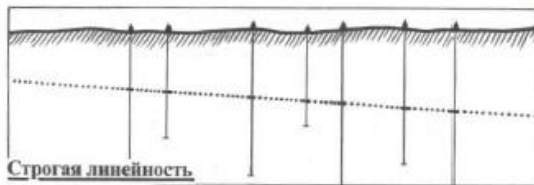
Фрактальна розмірність цього графіка дорівнює $D=3/2$ при наступним параметрах -

$$b > 1, 0 < w < 1, D = 2 + \log_b w, 1 < D < 2 .$$

Завдання 1.1. Для різноманітних значень w та b побудувати графіки функції Веєрштраса.

Корреляция геологических разрезов

слов

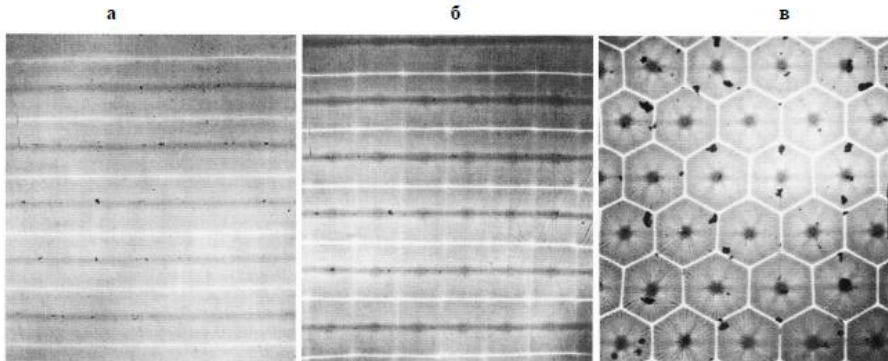


Конвекция Релея-Бенара при $R > R_{кр}$.

Число Релея $R = \frac{g\rho\gamma h^3 \Delta T}{\eta\chi}$

$$R_{кр} \sim 1000$$

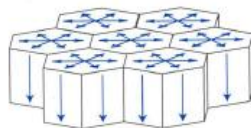
- мера количества теплоты, передаваемого посредством конвекции, к количеству теплоты, передаваемому кондуктивным (молекулярным) способом.



Результаты экспериментов по конвекции Релея-Бенара в слое вязкой жидкости.

(а) система двумерных валов; (б) ортогональные системы валов; (в) гексагональные ячейки, в которых горячее вещество поднимается в центре, а холодное опускается по углам ячейки.

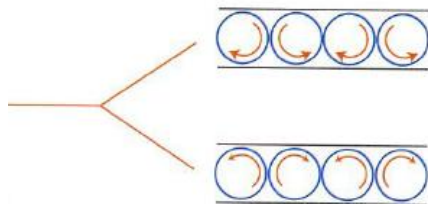
Выбор направления вращения валов в конвективных ячейках



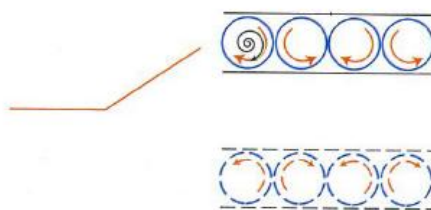
1 - гексагон

2 - гексагон

Направление движения жидкости в ячейках Бенара. Красными стрелками обозначены потоки более горячей жидкости, синими - более холодной. Высота ячеек сопоставима с их диаметром.



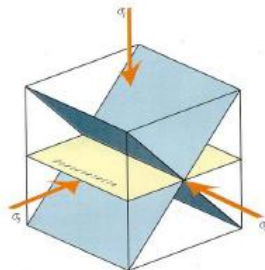
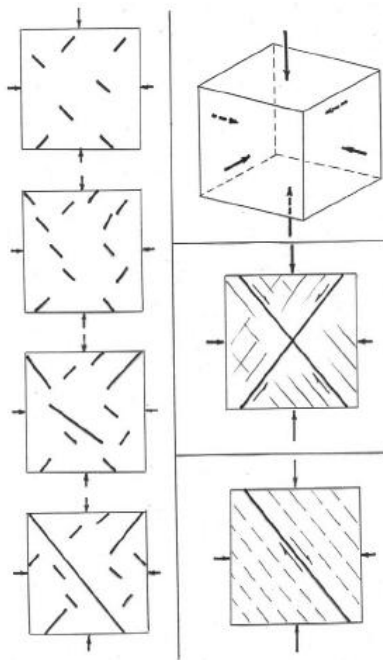
а



б

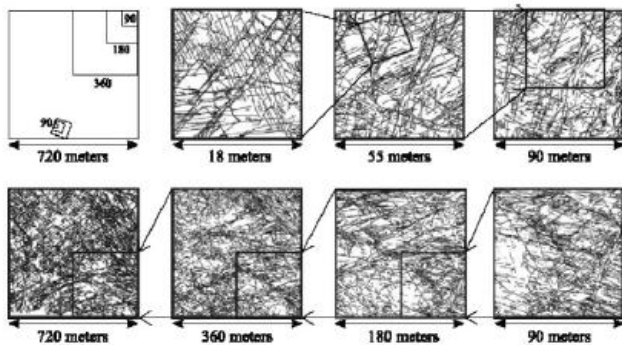
Бифуркационные диаграммы, иллюстрирующие множественность решений выше порога неустойчивости тепловой конвекции (а) и явление отбора за счет непрерывного внешнего воздействия (б).

Разрывообразование в блоке породы



Положение площадок максимальных касательных напряжений τ_{xy}

Свойства поля разрывных нарушений



$$n(l, L) = \alpha L^{D_M} l^{-D}, \quad (1)$$

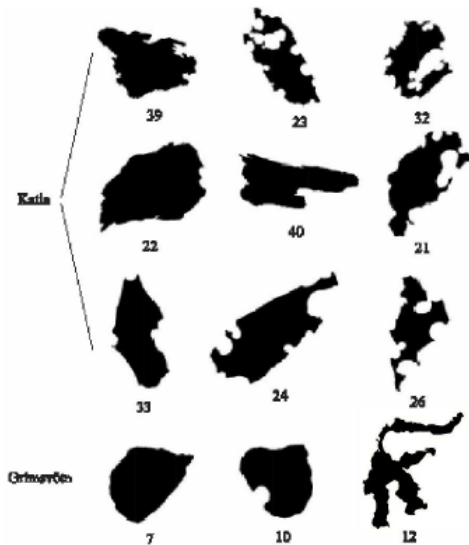
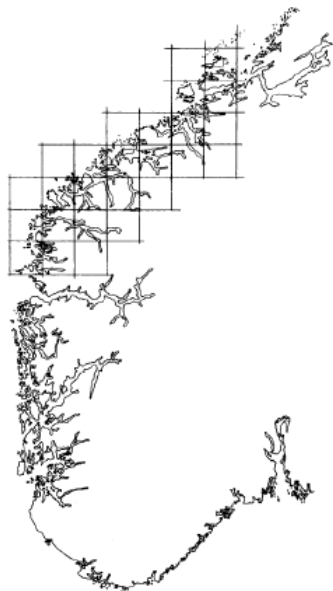


Складки

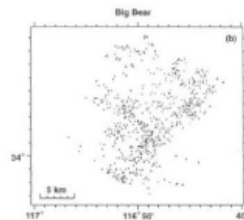
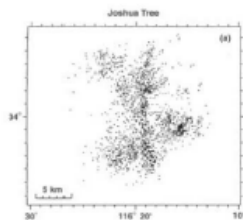
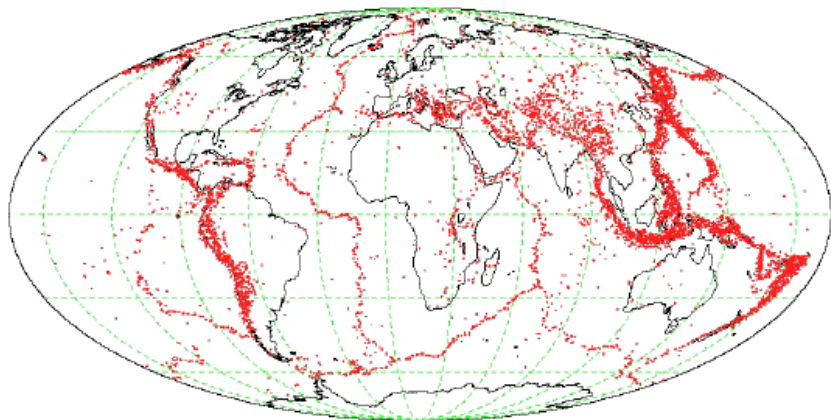


Дайки

Береговые линии.



Пространственное распределение землетрясений



1.1. ПОНЯТТЯ L-СИСТЕМ

Вперше поняття L-системи було введено угорським ботаніком Арістідом Ліндермаєром у 1968 році при вивченні ним розвитку простих багатоклітинних організмів.

Пізніше це поняття було поширеним на моделювання складних структур, що гілкуються (дерева, квіти), формальні мови, біологічні моделі відбору, фрактали та ін.

Найпростішою з L-систем є детерміновані контекстно-вільні D0L-системи.

Фактично L-система складається з побудованого в результаті ітераційного процесу слова і алгоритму його графічного відображення. Перша частина цілком відповідає вимогам лінгвістичного модулювання.

В нашому випадку слово складається з символів-команд, що належать алфавіту $A = \{F, b, +, -, [, .]\}$.

Алгоритм побудови слова полягає у наступному. До початкового слова $W^0 = W^0(F, b, +, -, [.])$ одночасно застосовуються породжувальні правила

$$F \rightarrow new F$$

$$b \rightarrow new b$$

тим самим породжуючи нове слово

$$W^1 = W^0(newF, newb, +, -, [.])$$

При цьому слід зазначити, що не припускається послідовна заміна символів на одному кроці ітераційного процесу.

Так не можна у слові Fb правилами

$$F \rightarrow b$$

$$b \rightarrow F$$

замінити спочатку F на b , отримавши bb , а надалі замінити b на F , отримавши FF . За нашим розумінням результат повинен бути bF . Ці особливості функціонування заміни в L-системах відмінюються від формальних граматики Хомського.

Надалі ітераційний процес повторюється, породжуючи послідовність слів W^0, W^1, W^2, \dots .

Легко бачити, що породжувана послідовність буде мати складну внутрішню структуру символів.

Наприклад, з початкового слова $W^0 = F$ та породжувачих правил $newF = F + F - - F + F$, $newb = b$ можна отримати наступні символльні конструкції:

$$W^0 = F,$$

$$W^1 = F + F - - F + F$$

$$W^2 = F + F - - F + F + F + F - - F + F - - F + F - - F + F + F + F - - F + F, \dots$$

Алгоритми графічного відображення символічних послідовностей називають черепаховою графікою. Якщо уявити черепаху у якості матеріальної точки на площині, а стан її трійкою (x, y, α) , де (x, y) - декартові координати на площині, а α - кут, відміряний від осі абсцис, в напрямку якого дивиться голова нашої Тортілли.

Процес відбувається наступним чином. Спочатку черепашка знаходиться в нульовому стані - (x_0, y_0, α_0) , наступний крок робить у напрямку свого погляду на відстань $d > 0$, при цьому повертаючи голову на постійний кут $\theta > 0$. При цьому наша черепашка читає слово зліва направо, й в залежності від прочитаного символу виконує наступні команди:

F — пролізти на крок вперед, промальовуючи при цьому свій слід (відрізок прямої);

d — пролізти на крок вперед, не промальовуючи свого сліду;

+ - повернути голову на кут θ ;

- - повернути голову на кут $-\theta$;

[- запам'ятати свій чинний стан (x, y, α) (команда гілкування);

] - повернутися в останній запам'ятований стан.

Продемонструємо роботу нашої Тортілли на прикладі символічних послідовностей W^0, W^1, W^2 , які ми отримали у вищенаведеному

прикладі (рис.1.3.1 та рис.1.3.2). При цьому маємо $\theta = \frac{\pi}{3}$.

Нижче приведений в узагальненому вигляді (без прив'язки до конкретної мови програмування) алгоритм реалізації L-системи та черепащачої графіки.

Вхід:	axiom;	слово-аксіома
	newF ;	слово-правило $F \mapsto \rightarrow newF$
	newb;	слово-правило $b \mapsto \rightarrow newb$
	level;	кількість ітерацій
	θ ;	кут повороту черепашки
Вихід:	W	результуюча послідовність символів

- 1: $W = axiom$; ініціалізація шуканої послідовності символів
- 2: повторюватиме level раз ів
- 3: Будуємо послідовність T, що отримуємо з W шляхом замін $F \mapsto \rightarrow newF$, $b \mapsto \rightarrow newb$;

4: $W = T;$

Інтерпретація символного ланцюжка W

5: $\alpha = 0;$

Ініціалізуємо початкове
спрямування черепашки

6: $(x_0, y_0) = (0, 0);$

Ініціалізуємо початкове
положення черепашки

7: $path = (x_0, y_0);$

Ініціалізуємо масив точок
ламаної шляху черепашки

8: $stack = \emptyset;$

Ініціалізуємо стек
гілкування

9: для $j = 1, \dots$, довжина ланцюжка
 W

10: якщо $W[j] = +$, то $\alpha = \alpha + \theta$;

11: якщо $W[j] = -$, то $\alpha = \alpha - \theta$;

12: якщо $W[j] = F$, то

13: $x = x_0 + \cos \alpha$;

14: $y = y_0 + \sin \alpha$;

15: $\text{path} = \text{path} \cup (x, y)$;

16: $(x_0, y_0) = (x, y)$;

17: якщо $W[j] = [$, то $\text{stack} = \text{stack} \cup$
 (x_0, y_0, α) ;

18: якщо $W[j] =]$, то

19: $(x_0, y_0, \alpha) = \text{stack}(-1)$;

дістаємо останній елемент із
стеку

20: Видаляємо останній елемент із стеку

21: З'єднуємо масив точок `path` ламаною лінією й виводимо на екран

Результат роботи програми на MAXIMA приведений на малюнку 1.2.

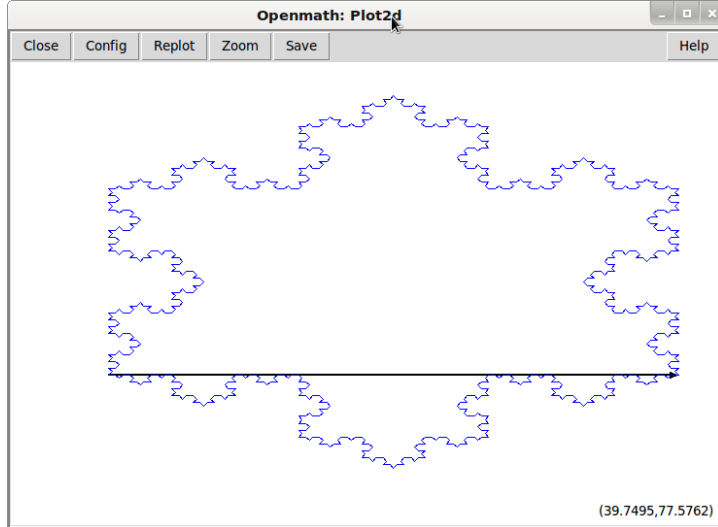


Рис.1.2. Результат роботи програми, реалізуючої L-систему та черепашкову графіку.

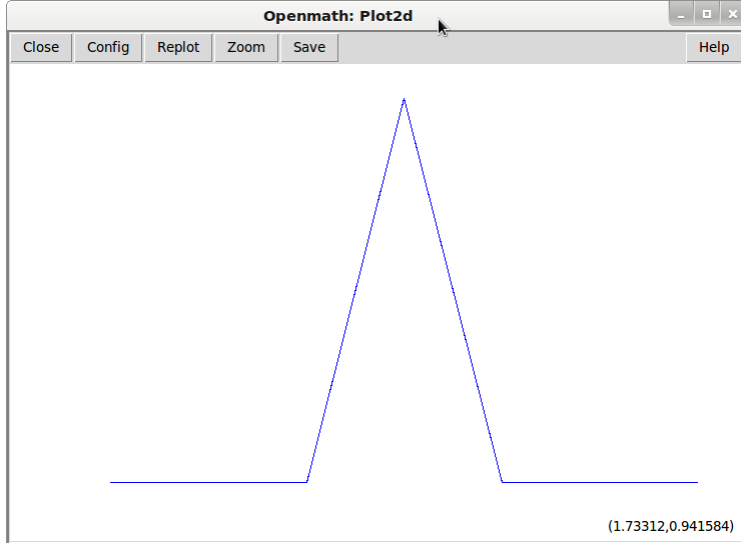


Рис.1.3.1. Відображення лінгвістичного ланцюжку
 $W^1 = F + F - - F + F$

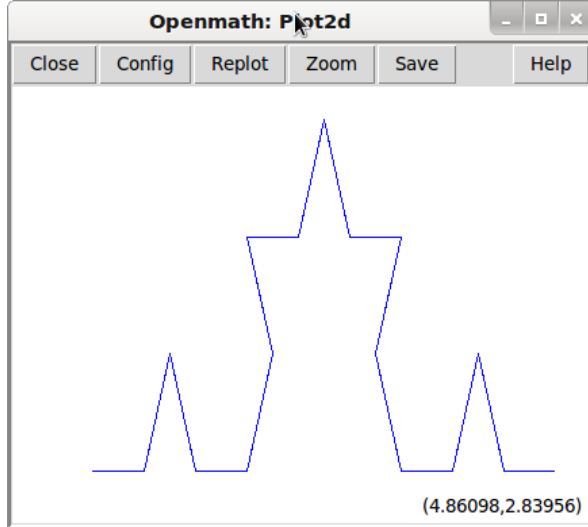


Рис.1.3.2. Відображення лінгвістичного ланцюжку
 $W^2 = F + F - - F + F + F + F - - F + F - - F + F - - F + F + F + F - - F + F, \dots$

1.2. КЛАСИФІКАЦІЯ L-СИСТЕМ

На малюнку рис.1.4 приведена загальна класифікація L-систем

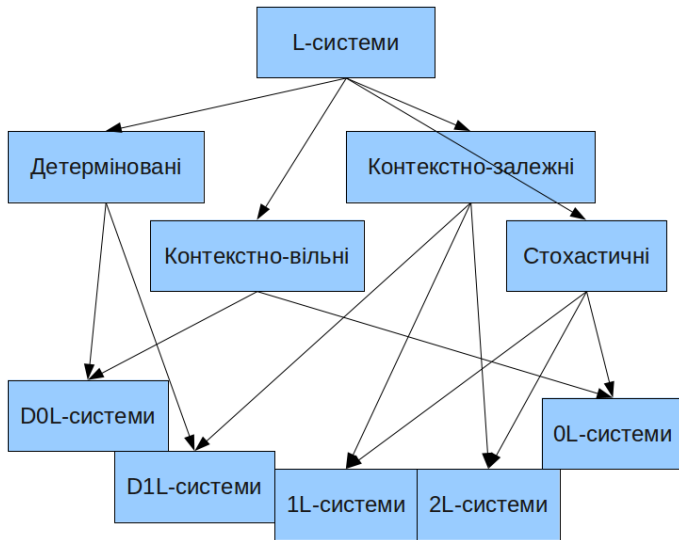


Рис.1.4. Загальна класифікація L-систем

Згідно ще одного варіанту класифікації виділяють наступні види L-систем:

- ▲ Детерміністичні контекстно-вільні системи
- ▲ Контекстно-залежні L-системи, де заміна символів залежить від найближчих сусідів
- ▲ Системи з гілкуванням з можливістю запам'ятовування стану черепашки та повернення до нього, а також вимальовування розгалужених структур
- ▲ Стохастичні системи з рандомізацією правил інтерпретації, а також з процесом побудови самої лінгвістичної послідовності
- ▲ Темпоральні системи з еволюціюванням правил з часом
- ▲ Параметричні системи, в яких з кожним символом в лінгвістичному ланцюжку пов'язаний набір параметрів й заміна цього символу відбувається при виконанні логічних умов, що мають відношення до параметрів

1.2.1. Системи з гілкуванням

Для реализации структур, похожих на ветви деревьев, используются парные команды ветвления [и] и массив-стек положений ветвления:

$$\begin{bmatrix} x_1 & y_1 & \alpha_1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & \alpha_N \end{bmatrix}.$$

Каждый раз при открытии ветви символом [в конец этого стека записывается текущее состояние (x_N, y_N, α_N) , а при закрытии ветви символом] черепашка возвращается на позицию перед ветвлением и из этого стека удаляется последний элемент. Стек необходим, потому что ветвления могут быть вложенными друг в друга, например F [+b+ [F+F]-b+F]+F. Примеры ветвящихся *L*-систем см. в Задачах 1.3, 1.5.

1.2.2. Контекстно-залежні L-системи

Каждый символ A может меняться в зависимости от своего *контекста* (окружения). В $2L$ -системах контекст двухсторонний: от окружающих его слева и справа символов (A_l и A_r) зависит порождающее правило:

$$A_l < A > A_r \mapsto newA.$$

Также используют дополнительные символы 0 и 1 , отвечающие за контекст, но не интерпретируемые черепашкой, и указывается, какие символы игнорируются при нахождении контекста: *ignore* : $+ - F$ означает, что, например, в слове

$$F\underline{0}FF\underline{1} + F\underline{0} - 1F$$

контекстом символа $A = 1$ в середине слова являются символы $A_l = 0$ и $A_r = 0$. Запись $* < A > * \mapsto newA$ означает, что символ A меняется независимо от контекста. Как правило, для символа F правила не задаются. Примеры таких $2L$ -систем см. в Таблице 1.3.

Если $2L$ -система содержит ветвления, то задание контекста несколько усложняется: близкие символы могут быть разделены длинной последовательностью ветвей. В одном из самых простых случаев дочерние ветви не пересекаются с контекстом материнских ветвей. Например, в слове

$$ABCD[EF][\underline{G}[HI[JKL]M]\underline{NOPQ}]$$

левым контекстом символа G является D , а правым контекстом — символ N .

В $1L$ -системах контекст односторонний: $A_l < A \mapsto newA$ или $A > A_r \mapsto newA$. Обобщением этих случаев являются IL -системы (k, l -системы), в которых контекст образуют k символов слева и l символов справа от A .

1.2.3. Стохастичні L-системи

Случайность в построении L -систем можно вводить на двух этапах (см. Задачи 1.1 и 1.2):

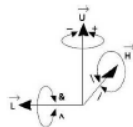
- При построении слова вместо одного правила для символа F используется N правил $newF_1, newF_2, \dots, newF_N$, выбираемых на каждом шаге итерационного процесса с соответствующими вероятностями p_1, p_2, \dots, p_N ($\sum_{n=1}^N p_n = 1$). Обозначение:

$$\begin{aligned} p_1: F &\xrightarrow{p_1} newF_1, \\ &\vdots \\ p_N: F &\xrightarrow{p_N} newF_N. \end{aligned}$$

- При интерпретации слова можно менять длину шага черепашки и угол поворота ее головы случайным образом в соответствии с некоторыми распределениями вероятностей.

1.2.4. 3D L-систем

Кроме плоского случая (2D), рассматриваются и 3D реализации L-систем. Ориентация черепашки в пространстве задается ортонормированным правым базисом \vec{H} , \vec{L} и \vec{U} , в котором \vec{H} (heading) — направление *взгляда* черепашки, \vec{L} (left) — направление *налево*, \vec{U} (up) — *вверх*. Вместо одной пары команд поворота на постоянный угол вводятся три: +, - (повороты вокруг оси \vec{U}), ^, & (повороты вокруг оси \vec{L}), /, \ (повороты вокруг оси \vec{H}).



1.2.5. L-системы с двумя символами

Для удобства построения некоторых систем вместо одного символа шага F полезно ввести два символа F_x и F_y . Например:

$$\begin{aligned} \text{axiom} &= F_x, \\ F_x &\mapsto F_x + F_y+, \\ F_y &\mapsto -F_x - F_y. \end{aligned}$$

Однако тот же результат можно получить, если символ шага F будет один, но добавятся еще два не интерпретируемых черепашкой символа X и Y :

$$\begin{aligned} \text{axiom} &= FX, \\ X &\mapsto X + YF+, \\ Y &\mapsto -FX - Y. \end{aligned}$$

Примеры таких L -систем см. в Таблице 1.2. См. также Задачу 1.9.

ЛИТЕРАТУРА

[1e] <http://maxima.sourceforge.net/ru/>

Официальная страница проекта Maxima на русском языке (однако, она редко обновляется, рекомендуется пользоваться английской версией страницы).

[2e] <http://sourceforge.net/projects/maxima/files/>

Здесь можно скачать установочный файл для Maxima.

[3e] <http://maxima.sourceforge.net/ru/documentation.html>

Подборка руководств по Maxima с официального сайта, в том числе и на русском языке.

[4e] <http://andrejv.github.com/wxmaxima/help.html>

Коллекция руководств по Maxima.

[5e] http://www.uneex.ru/static/MethodBooks_Maxima/Maxima.pdf

Пособие Н.А. Стахина .Основы работы с системой аналитических (символьных) вычислений Maxima. на русском языке.

[6e] <http://www3.msiu.ru/~lao4/maxima.pdf>

Методическое пособие с неофициального сайта МГИУ www3.msiu.ru.

[7e] <http://euler.us.es/~renato/clases/maxima/manualesPDF/MaximaByExamples/mbe5qdraw.pdf>

Документация пакета Edwin L. Woollett qdraw.

[8e] <http://www.csulb.edu/~woollett/>

Сайт профессора Edwin L. Woollett с очень полезной информацией о Maxima.

[9e] <http://www.csulb.edu/~woollett/mbe2plotfit.pdf>

Тutorial от профессора Edwin L. Woollett о выводе графики и работе с файлами (см. также другие его туториалы по предыдущей ссылке).

[10e] <http://en.wikibooks.org/wiki/Maxima>

Полезная информация о Maxima в открытой энциклопедии. Есть примеры программирования фракталов.

[11e] <http://www.telefonica.net/web2/biomates/maxima/sound/index.html>

Пакет работы со звуком `sound.lisp`.

[12e] <http://maxima.cvs.sourceforge.net/viewvc/maxima/maxima/share/contrib/fractals/>

Пакет для построения простейших фракталов `fractals.mac`. На этом же сайте есть много других пакетов для Maxima.

[13e] <http://www.imagemagick.org/script/download.php>

Отличная бесплатная программа Image Magick для работы с изображениями из командной строки.

[14e] http://www.biologie.uni-hamburg.de/b-online/e28_3/lsys.html

L-системы.

[15e] <http://algorithmicbotany.org/>

L-системы. Сайт профессора Пршемыслава Прузинкевича.

[16e] <http://algorithmicbotany.org/papers/#abop>

Книга Прузинкевича в открытом доступе.

[17e] <http://mathworld.wolfram.com/LindenmayerSystem.html>

Черпаем вдохновение в готовых демонстрациях с сайта mathworld.wolfram.com.

[18e] http://www.austromath.at/daten/maxima/zusatz/Graphics_with_Maxima.pdf

Хорошая документация по использованию графики в Maxima.

[19e] <http://riotorto.users.sourceforge.net/gnuplot/animation/>

Примеры создания анимации средствами Maxima.

[20e] <http://riotorto.users.sourceforge.net/gnuplot/>

Хорошая подборка примеров использования графики от создателя пакета draw
Mario Rodr'iguez Riotorto.

[21e] [http://en.wikibooks.org/wiki/Fractals/Iterations_in_the_complex_plane/
Julia_set](http://en.wikibooks.org/wiki/Fractals/Iterations_in_the_complex_plane/Julia_set)

Алгоритмы построения и раскрашивания Множеств Жюлиа.

[22e] <http://www.fraktal.republika.pl/iim.html>

Реализации метода обратных итераций для построения множеств Жюлиа.

[23e] <http://www.fraktal.republika.pl>

Различные алгоритмы для изучения множеств Жюлиа, множества
Мандельброта и др., а также полезные ссылки.