

4. Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation. Winter 1989, vol. 1, no. 4, P. 541-551.

5. S. Ioffe, C. Szegedy Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. – [Электронный ресурс]// proceedings IEEE Conf. on Computer Vision and Pattern Recognition. 2001. 9 p. URL: <https://arxiv.org/pdf/1502.03167.pdf> (date of the application 21.12.2019).

6. REG.RU: Как начать работу с Keras, Deep Learning и Python: сайт. – URL: <https://www.reg.ru/blog/keras/> (дата обращения: 21.12.2019). – Текст: электронный.

7. Акулич И.Л. Математическое программирование в примерах и задачах. М.: Высшая школа, 1986. 310 с.

## АНАЛИЗ И ПОСТРОЕНИЕ ФРАКТАЛОВ В РАЗЛИЧНЫХ ПРОГРАММНЫХ СРЕДАХ

Грищенко А.Ю., Фоменко Л.Н.

ФГБОУ ВО «Донской Государственный  
Технический Университет», Ростов-на-Дону,  
e-mail: [reception@donstu.ru](mailto:reception@donstu.ru)

Фрактальная геометрия одна из самых молодых областей математики. С помощью фрактальных алгоритмов возможно описание различных природных процессов, обладающих самоподобием (ветка дерева подобна всему дереву, при изменении масштаба береговая линия остается похожей на себя и др.) [1].

Фрактал (лат. *fractus* – дроблённый, сломанный, разбитый) – объект дробной размерности, обладающий свойством фрактального самоподобия (скейлинга) (например, кривая Коха, ковер Серпинского, траектория броуновской частицы и т.д.). В простых случаях небольшая часть фрактала содержит информацию о всем фрактале. Само понятие фрактал придумал ученый-математик Бенуа Мандельброт и ввел его в использование в 1975 году [2].

Впервые фрактал был построен в 1883 году Георгом Кантором. В то время он был определен как математический монстр. Кантор взял отрезок и разделил его на три части. Среднюю часть он вырезал, после чего оставались два крайних отрезка. Теперь из каждого оставшихся отрезков он вновь вырезал среднюю часть и так до бесконечности (рис. 1).

Отцом фрактальной геометрии и создателем понятия фрактал является американский, выросший во Франции математик – Бенуа Мандельброт. В 1967 году он опубликовал свою работу «Какова длина побережья Великобритании?», и именно она считается отправной точкой в цикле его работ по теории фракталов. В 1977 году вышла книга Мандельброта «Fractals: Form, Chance and Dimension» [3].

В нашей работе мы обратились к одному из самых известных фракталов – множеству Мандельброта [3]. Алгоритм построения такого множества основан на итеративном выражении (1):

$$Z_{n+1} = Z_n^2 + C. \quad (1)$$

Здесь  $Z_{n+1}$  и  $Z_n$ ,  $C$  – комплексные переменные. Длительность итерационного процесса зависит, сойдется ли  $Z_n$  к какой-либо точке окруж-

ности радиуса два и центром в начале координат, тогда эта точка  $C$  закрашивается в черный цвет. Точки, уходящие в бесконечность за конечное число итераций, образуют белый фон. Те точки, которые не уходят в бесконечность при конечном числе итераций, принадлежат границе множества, где и возникают сложные структуры. Выбирая в качестве фазового пространства двумерное, получим фазовый портрет итерационного процесса такой системы (рис. 2).

Фрактальный рисунок не имеет идентичных элементов, но обладает подобностью в любом масштабе. Построить такое изображение с высокой степенью детализации вручную ранее было просто невозможно, на это требовалось огромное количество вычислений.

На данный момент времени существует огромное количество программ для построения фракталов и генерации сложных неевклидовых объектов, например искусственных гор, облаков и др. [4]. Благодаря несложным формулам, фракталы можно легко запрограммировать.

В данной статье для создания и качественной визуализации данных мы будем применять инженерное математическое программное обеспечение Mathcad и высокоуровневый язык программирования Python с библиотекой Matplotlib. Проведем сравнительный анализ удобства работы с фракталами в этих средах.

Для построения фрактала в среде Mathcad будем использовать формулу множества Мандельброта (1) и встроенные функции Mathcad (рис. 3). Параметры width (ширина изображения в пикселях) и height (высота изображения) выбираются по желанию пользователя. Значение параметра iterations (количество итераций) так же выставляет пользователь (в нашем случае 29), но не более 100, т.к. при увеличении количества итераций изображение не изменяется ввиду ограниченных вычислительных способностей среды Mathcad.  $x1$ ,  $x2$ ,  $y1$ ,  $y2$  – координаты на плоскости, внутри которых мы будем строить множество Мандельброта,  $C_{ij}$  – вычисляемая матрица комплексных чисел.

Среда Mathcad проста в применении, не требует углубленных знаний информатики и языков программирования. Среди её минусов возможно отметить довольно запутанный и неудобный интерфейс программы.

Для построения и визуализации множества Мандельброта с использованием возможностей языка программирования Python на основании формулы (1) определим родительскую функцию (в нашем случае она названа mandel). Для размерности изображения необходима еще одна функция, определяемая пользователем (в нашем случае create\_fractal). Для удобства визуализации графика подключим библиотеку Matplotlib. Применяя цикл с параметром, программа формирует график с поэтапным окрашиванием изображения с каждой новой итерацией.



Рис. 1. Канторово множество

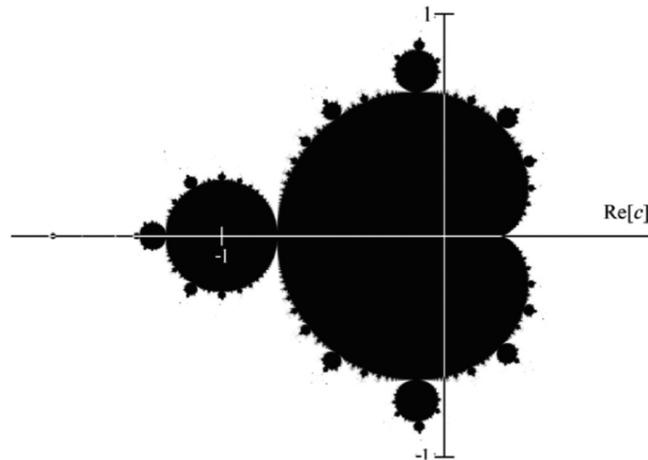


Рис. 2. Множество Мандельброта

```

x1 := -1.5    y1 := -2.
x2 := 1.5    y2 := 1.0

width := 299    iterations := 25
height := 299

i := 0..width    j := 0..height
Ci,j := [x1 +  $\frac{j \cdot (x2 - x1)}{\text{height}}$  -  $\sqrt{-1}$  + [y1 +  $\frac{i \cdot (y2 - y1)}{\text{width}}$ ]
Zi,j := | Zi,j ← 0
        | counter ← 0
        | while counter < iterations
        |   | Zi,j ← (Zi,j)2 + Ci,j
        |   | counter ← counter + 1
        |   | break if |Zi,j| > 2
        | if (0, counter < iterations, counter ·  $\frac{\text{width}}{\text{iterations}}$ )

WRITEBMP("C:\Users\Александр\Desktop\Z.bmp", Z) = 0

```

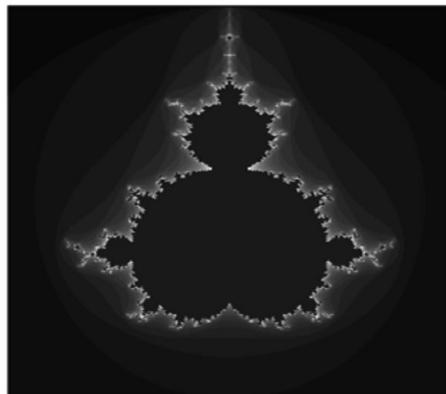


Рис. 3. Рабочий лист Mathcad с выводом результата

Ниже приведен код программы в Python 3.7 (рис. 4) и полученный график (рис. 5).

Отметим, что при программировании множества Мандельброта в среде Python нам удалось создать лаконичный код с подключением

одной библиотеки. Учитывая, что возможности языка программирования Python динамично развиваются, в перспективе возможно исследование поведения графика при масштабировании и детализации изображения.

```

from pylab import imshow, show

thisISAVariable = 5

max_iters = 20

def mandel(x, y, max_iters):
    c = complex(x, y)
    z = 0.0j
    for i in range(max_iters):
        z = z * z + c
        if (z.real * z.real + z.imag * z.imag) >= 4:
            return i
    return max_iters

def create_fractal(min_x, max_x, min_y, max_y, image, iters):
    height = image.shape[0]
    width = image.shape[1]

    pixel_size_x = (max_x - min_x) / width
    pixel_size_y = (max_y - min_y) / height

    for x in range(width):
        real = min_x + x * pixel_size_x
        for y in range(height):
            imag = min_y + y * pixel_size_y
            color = mandel(real, imag, iters)
            image[y, x] = color

image = np.zeros((1024, 1536), dtype=np.uint8)
create_fractal(-2.0, 1.0, -1.0, 1.0, image, max_iters)

imshow(image)
show()

```

Рис. 4. Код программы в Python 3.7

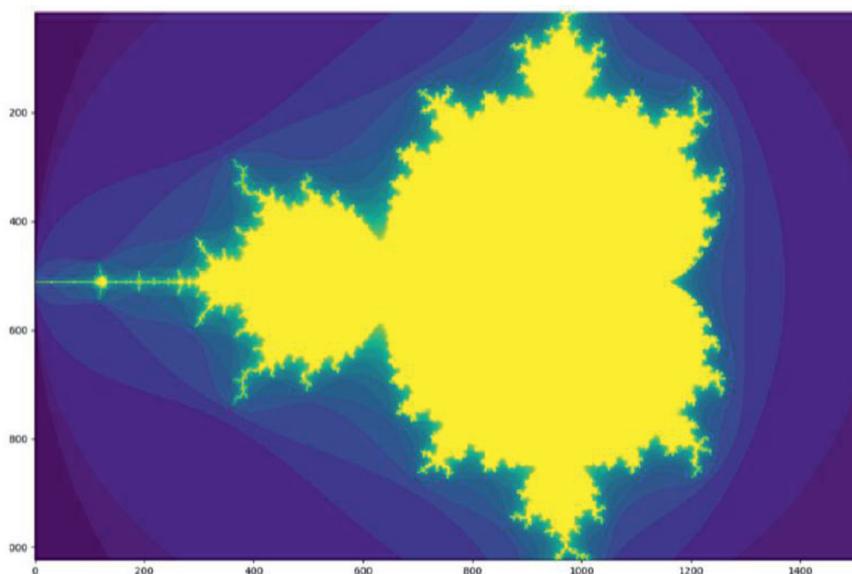


Рис. 5. Результат работы в Python

**Список литературы**

1. Секованов В.С. Что такое фрактальная геометрия? – Москва, 2016. Сер. 75, Выпуск 114 Синергетика: от прошлого к будущему. Шедевры научно-популярной литературы (физика).
2. Савченко В.Н., Смагин В.П. Начала современного естествознания. Тезаурус. – Ростов-на-Дону, 2006.
3. Мандельброт Б. Fractals: Form, Chance and Dimension – United States, 1977.
4. Мандельброт Б. Фракталы и хаос. Множество Мандельброта и другие чудеса // Бенуа Мандельброт. – Ижевск: НИЦ «Регулярная и хаотическая динамика», 2009.

**АЛЬТЕРНАТИВНЫЕ ВИДА ТОПЛИВА  
ДЛЯ ВОДОГРЕЙНЫХ  
ОТОПИТЕЛЬНЫХ КОТЛОВ**

Зольников А.С., Вяхирев К.А.

*ФГБОУ ННГАСУ, Нижний Новгород,  
e-mail: srec@nngasu.ru*

В данном тезисе рассмотрено альтернативное топливо – отработанные масла. Представлена информация об экономичности и экологичности этого топлива. Показан принцип работы котла, работающего на отработанном масле. Выявлены преимущества и недостатки таких котлов.

С каждым годом цены на сырую нефть и нефтепродукты растут, поэтому возникает потребность разработки и интеграции новых способов и технических решений результативного применения топлива, тепловой энергии и вторичных энергоресурсов в промышленности.

В настоящее время актуально прибегать к поискам альтернативного вида топлива для водогрейных отопительных котлов и это дает неожиданные результаты. Все, что утилизировалось раньше, сейчас можно использовать как энергоноситель. Отработанное масло – это смазывающая жидкость, проработавшая свой срок и утратившая свои первоначальные свойства. Тем не менее, необходимая степень горючести будет поддерживаться оставшимися в нем спиртами и этиленгликолем. При сжигании 1 л отработанного масла выделяется около 11 кВт тепловой энергии, что можно сравнить с дизельным топливом.

К сожалению, на данный момент в России отсутствует единая система сбора, переработки и использования отработанного масла. Несомненно, предприятия, перерабатывающие отработанные масла существуют в нашей стране, но из собранного масла они восстанавливают солидол и другие низкосортные жидкости, которые не имеют популярности. Однако, весьма перспективно использовать «отработку» для отопления, так как потенциал топливной базы в России составляет до 500 млн. т/год и более.

На повторное использование идет меньше половины (40–48%): 14–15% из них идет на восстановление, остальное уже используется как топливо. Обогрев промышленных объектов – это одно из наиболее многообещающих на-

правлений утилизации использованного масла. За счет отказа от потребления традиционного газового или жидкого топлива, расходы на утилизацию отработанного масла как опасного промышленного отхода и другое, предприниматель сможет существенно снизить затраты. Таким образом, нагрузка на окружающую среду снизится благодаря сжиганию отработанного масла с помощью оборудования, которое безопасно и высокотехнологично, с точки зрения экологии, по сравнению с неконтролируемой «утилизацией».

Экономическая выгода очевидна при использовании «своего» отработанного масла, стоимость которого равна нулю минус затраты, связанные с утилизацией.

**Список литературы**

1. Авласевич А.И. Использование отработанного масла в качестве топлива / А.И. Авласевич, И.Б. Оленев, А.С. Климов // Вестник Красноярского государственного аграрного университета. – Красноярск, 2012. – Вып. № 1. – С. 147–153.
2. Григорьев А.В. Отопление на отработанном масле / А.В. Григорьев // Промышленные и отопительные котельные и мини-ТЭЦ. – 2011. – Вып. № 2(7).
3. Шашкин П.И. «Регенерация отработанных нефтяных масел».

**КВАЛИМЕТРИЧЕСКАЯ ОЦЕНКА  
УРОВНЯ КАЧЕСТВА РУБЛЕННОГО  
МЯСНОГО ПОЛУФАБРИКАТА  
«ЛЮЛЯ-КЕБАБ»**

Котова А.Д., Дворянинова О.П., Пегина А.Н.

*ФГБОУ ВО «Воронежский государственный  
университет инженерных технологий»,  
Воронеж, e-mail: adkotova98@mail.ru*

В настоящее время качество выпускаемой продукции по праву можно отнести к важнейшим критериям деятельности любого предприятия. От качества продукции зависит конкурентоспособность, объем выпускаемой продукции, цена, прибыль и возможность развития.

Для анализа качества выпускаемой продукции была проведена квалиметрическая оценка качества мясных полуфабрикатов пяти производителей (Производитель 1, Производитель 2, Производитель 3, Производитель 4, Производитель 5), которая состояла из: установления требований потребителей; определения номенклатуры показателей качества выбранных мясных полуфабрикатов; определения значений свойств показателей качества; выбора базового образца; оценки уровня качества изделий.

Для определения номенклатуры показателей качества было построено «дерево свойств», которое создавалось по результатам опроса, взятого с официального источника Роскачество. По полученным данным были выявлены наиболее важные для потребителей показатели качества такие, как внешний вид, консистенция, запах и вкус, массовая доля белка, массовая доля жира, массовая доля хлористого натрия, массовая доля общего фосфора.