

УДК 519.6 (076.5)(075.8)

ББК 22.19я73

Г60

Рекомендовано ученым советом
механико-математического факультета
14 июня 2011г., протокол №8

Р е ц е н з е н т ы:

доктор физико-математических наук,
профессор *А. П. Садовский*;
доктор технических наук,
профессор *В. А. Липницкий*

Голубева, Л.Л.

Г60 Компьютерная математика. Символьный пакет *Mathematica* : лаб.
практикум. В 2 ч. Ч. 1. / Л. Л. Голубева, А. Э. Малевич,
Н. Л. Щеглова. – Минск: БГУ, 2012. – 231 с.

ISBN 978-985-518-783-8

В практикуме изложены основные принципы работы в символьном математическом пакете *Mathematica*, содержится необходимые теоретические сведения, примеры выполнения заданий, а также упражнения для самостоятельной работы и перечень контрольных вопросов.

Практикум предназначен для студентов механико-математического факультета БГУ очной и заочной форм обучения, а также может быть полезен для студентов физико-математических и инженерных специальностей.

УДК 519.6 (076.5)(075.8)

ББК 22.19я73

ISBN 978-985-518-783-8 (ч.1)

ISBN 978-985-518-782-1

© Голубева Л. Л., Малевича. А. Э.,
Щеглова Н. Л., 2012

© БГУ, 2012

ЛАБОРАТОРНАЯ РАБОТА 5 (ЧАСТЬ 3)

Задание 5 (часть 3)

Создайте мультфильм, изображающий замкнутую ломаную последовательно поступающих точек плоскости. В кадре, помимо графического образа объекта, отобразите его свойства: количество точек, их координаты, периметр замкнутой ломаной, диаметр множества ее вершин.

5.11. Виртуальная модель замкнутой ломаной

Задание 5.9

Построить виртуальную модель замкнутой ломаной.

Выполнение задания 5.9

Чтобы построить виртуальную модель замкнутой ломаной, обратимся к определению понятия ломаная линия. Это позволит выделить информацию, которая необходима для представления объекта ломаная линия в *Mathematica*.

Ломаная линия – это объединение отрезков, в котором конец каждого отрезка (кроме, возможно, последнего) является началом следующего отрезка, причем отрезки, имеющие общий конец, не лежат на одной прямой. Ломаная линия замкнута, если конец последнего отрезка совпадает с началом первого отрезка. Концы отрезков называют вершинами ломаной, отрезки, составляющие ломаную, – звеньями ломаной, отрезки, имеющие общий конец – смежными звеньями ломаной.

Из определения следует, что для представления объекта замкнутая ломаная последовательно поступающих точек плоскости нужно указать список $P_0, P_1, \dots, P_n$ точек-координат вершин ломаной. Кроме того, согласно условию задачи, в представлении следует отразить следующие свойства замкнутой ломаной: количество вершин, сумму длин звеньев – периметр, диаметр множества ее вершин.

В *Mathematica* все есть выражение, поэтому представим объект замкнутая ломаная в виде выражения

BreakLine[Counter, Verticies, Perimeter, Diameter], (5.2)

где **Counter** – количество вершин, **Verticies** – список координат вершин, **Perimeter** – периметр и **Diameter** – диаметр замкнутой ломаной.

Поскольку по условию задачи точки в замкнутую ломаную поступают последовательно, то ломаная линия может содержать ноль, одну (нерегулярные случаи) либо много (регулярный случай) точек.

Согласно определению, не каждая вновь поступающая точка может быть вершиной замкнутой ломаной. Пусть на k -м шаге замкнутая ломаная содержит n вершин $P_0, P_1, \dots, P_n$. При поступлении новой точки P_{n+1} следует проверить, может ли она быть следующей вершиной новой ломаной. А именно, точка P_{n+1} не должна лежать на прямой, проходящей через вершины $P_{n-1}P_n$.

Функция **PointOnLine[{a,b},c]** тестирует принадлежность точки C прямой, проходящей через две различные точки a и b . Каждая точка указывается списком своих координат. Функция **PointOnLine** возвращает значение **True**, если точка c принадлежит прямой ab и **False** в противном случае.

PointOnLine[{a_, b_}, p_] := Det[{b - a, p - a}] == 0

PointOnLine[{{-1, -1}, {5, 5}}, {7, 7}] **True**

PointOnLine[{{-1, -1}, {5, 5}}, {1, 7}] **False**

Обсудим далее, как вычислять следующие свойства замкнутой ломаной:

- количество вершин;
- сумму длин отрезков, ее составляющих;
- диаметр множества вершин.

Для нахождения длины звена ломаной и диаметра множества ее вершин понадобится вычислять расстояние между двумя точками. Эту операцию оформим в виде отдельной функции **Distance**:

Distance[A : {_, _}, B : {_, _}] := Sqrt[#.#] &[A - B]

В задачах вычисления свойств нового объекта возможны разные стратегии:

- использовать ранее вычисленное свойство старого объекта;
- использовать данные только этого нового объекта.

Выберите оптимальную на Ваш взгляд стратегию вычисления количества вершин, периметра, диаметра замкнутой ломаной, аргументируйте Ваш выбор, опишите его реализацию.

5.12. Графическое представление замкнутой ломаной

Задание 5.10

Постройте функцию, вычисляющую графическое представление замкнутой ломаной.

Выполнение задания 5.10

Построим функцию **Show**, которая отображает замкнутую ломаную вида (5.2). Эта функция понадобится нам при тестировании работы других функций и для создания мультфильма. Функцию **Show** зададим на множестве объектов с головой **BreakLine**. Правило, которое определяет функция **Show**, ассоциируем с символом **BreakLine**:

BreakLine/:Show[BL_BreakLine]:=FunctionBody (5.3)

Функция **Show** будет отображать:

- вершины замкнутой ломаной;
- звенья замкнутой ломаной;
- порядковые номера последовательно поступающих точек-вершин;
- структуру и данные объекта **BreakLine**.

Прежде чем построить описанную выше функцию **Show**, последовательно обучимся строить указанные объекты: вначале отобразим вершины, затем добавим звенья, и, наконец, для каждой точки-вершины ломаной линии укажем ее порядковый номер поступления.

Сгенерируем объект ломаная линия вида (5.2), задавая координаты точек произвольным образом.

Size=50;

P=RandomInteger[{0,Size},RandomInteger[{3, 15}]]

Пусть на **k**-м шаге добавления точки ломаная линия **BL** имеет вид

BL = BreakLine[Length[P], P, Perimeter, Diameter]

```
BreakLine[9, {{37, 4}, {46, 18}, {42, 6}, {50, 15},
  {21, 30}, {47, 8}, {6, 37}, {4, 11}, {3, 10}},
  Perimeter, Diameter]
```

С помощью функции `RandomInteger` мы сгенерировали объект `BreakLine`, не вычисляя пока его периметр и диаметр.

Построим вершины замкнутой ломаной (рис. 5.11, а), используем графический примитив `Point`.

```
BL[[2]]
```

```
{{37, 4}, {46, 18}, {42, 6}, {50, 15},
  {21, 30}, {47, 8}, {6, 37}, {4, 11}, {3, 10}}
```

```
Point[#]&/@BL[[2]]
```

```
{Point[{37, 4}], Point[{46, 18}], Point[{42, 6}],
  Point[{50, 15}], Point[{21, 30}], Point[{47, 8}],
  Point[{6, 37}], Point[{4, 11}], Point[{3, 10}]}
```

Fig1 =

```
Graphics[{AbsolutePointSize[15], Hue[.5, .8, 1],
  Point[#] & /@ BL[[2]], AspectRatio → Automatic,
  PlotRange → {{-1, Size + 1}, {-1, Size + 1}}, Axes → True]
```

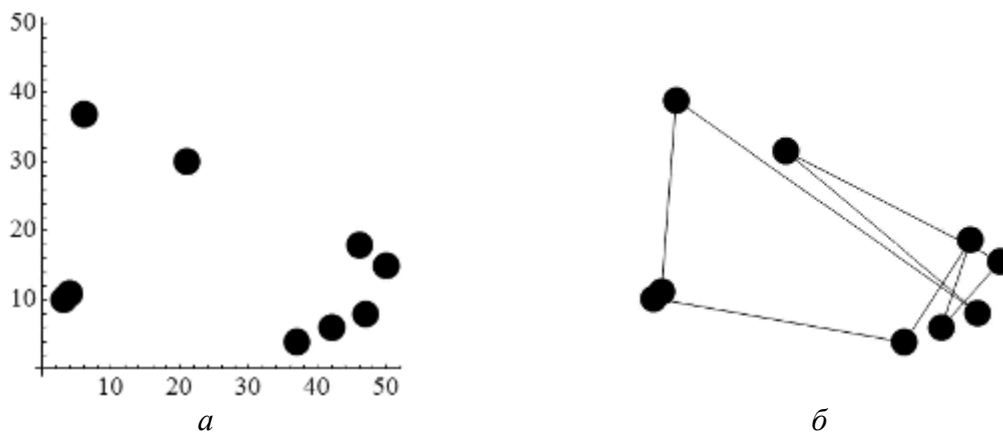


Рис. 5.1. Построение вершин и звеньев ломаной линии

Далее построим звенья ломаной линии `BL` (рис. 5.11, б), используем графический примитив `Line`. Чтобы замкнуть звенья, добавим первую вершину в конец списка координат вершин:

```
{Hue[0.051], Append[BL[[2], BL[[2, 1]]] // Line}
```

```
{Hue[0.051],
  Line[{{37, 4}, {46, 18}, {42, 6}, {50, 15}, {21, 30},
        {47, 8}, {6, 37}, {4, 11}, {3, 10}, {37, 4}}]}
```

Fig2 =

```
Graphics[
  {{AbsolutePointSize[15], Hue[.5, .8, 1], Point /@ BL[[2]]},
   {Hue[.5, .7, .4], Append[BL[[2]], BL[[2, 1]] // Line}},
  AspectRatio → Automatic,
  PlotRange → {{-1, Size + 1}, {-1, Size + 1}}]
```

Подпишем вершины ломаной линии BL в порядке их поступления. Используем функцию `MapIndexed[func, expr]` и графический примитив `Text`.

Аргумент `func` функции `MapIndexed` является функцией двух переменных, при вычислении функции `MapIndexed` на место первого #1 аргумента функции `func` подставляется очередное подвыражение выражения `expr`, на место второго #2 аргумента `func` – позиция этого подвыражения в структуре выражения `expr`.

По умолчанию, когда `MapIndexed` вызывается только с двумя аргументами, в выражении `expr` обрабатываются только подвыражения первого уровня. Например,

```
MapIndexed[(#1^#2) &, {a, b, c}]           {{a}, {b2}, {c3}}
```

```
MapIndexed[(#1^First@#2) &, {a, b, c}] {a, b2, c3}
```

Выражение `BL[[2]]` является списком, каждый элемент которого – список, содержащий координаты одной вершины ломаной. Таким образом, для отображения номера вершины удобно использовать ее позицию #2 в списке `BL[[2]]`.

```
MapIndexed[Text[First@#2, #1] &, BL[[2]]]
```

```
{Text[1, {37, 4}], Text[2, {46, 18}], Text[3, {42, 6}],
  Text[4, {50, 15}], Text[5, {21, 30}], Text[6, {47, 8}],
  Text[7, {6, 37}], Text[8, {4, 11}], Text[9, {3, 10}]}
```

Подготовим отображение и точки, и ее номера:

```
MapIndexed[{{Point[#1]}, Text[First@#2, #1]} &, BL[[2]]]
```

```

{{{Point[{37, 4}]}}, Text[1, {37, 4}]}},
 {{Point[{46, 18}]}}, Text[2, {46, 18}]}},
 {{Point[{42, 6}]}}, Text[3, {42, 6}]}},
 {{Point[{50, 15}]}}, Text[4, {50, 15}]}},
 {{Point[{21, 30}]}}, Text[5, {21, 30}]}},
 {{Point[{47, 8}]}}, Text[6, {47, 8}]}},
 {{Point[{6, 37}]}}, Text[7, {6, 37}]}},
 {{Point[{4, 11}]}}, Text[8, {4, 11}]}},
 {{Point[{3, 10}]}}, Text[9, {3, 10}]}},

```

Построим ломаную линию (рис. 5.12), отмечая точками ее вершины и нумеруя их:

```

Fig3 = Graphics[{AbsolutePointSize[15],
MapIndexed[
  {{{Hue[.9, .4, 1], Point[#1]}}, Text[First@#2, #1]} &,
  BL[[2]], Hue[.7, .8, 1], Append[BL[[2]], BL[[2, 1]] // Line},
AspectRatio → Automatic,
PlotRange → {{-1, Size + 1}, {-1, Size + 1}}]

```

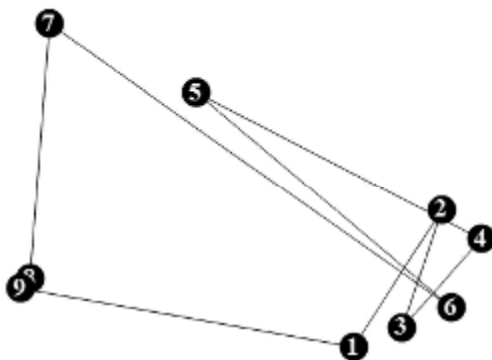


Рис. 5.2. Ломаная линия с занумерованными вершинами

Обобщая все выполненные ранее построения, построим функцию Show вида (5.3). Ее определим на выражениях, имеющих структуру (5.2), вводимое правило ассоциируем с символом BreakLine:

```

ClearAll[BreakLine]
BreakLine /: Show[BL_BreakLine] :=
  Graphics[{AbsolutePointSize[15], Hue[.9, .4, .1],
    Point[#] & /@ BL[[2]], Hue[.7, .8, 1],
    Line[Append[BL[[2]], BL[[2, 1]]],
    MapIndexed[{Text[First@#2, #1] &, BL[[2]]}],
  AspectRatio → Automatic, Axes → True,
  PlotRange →
    {{-1, Max[Table[BL[[2, i, 1]], {i, 1, Length[BL[[2]]}]] + 1},
     {-1, Max[Table[BL[[2, i, 2]], {i, 1, Length[BL[[2]]}]] + 1}}]

```

5.13. Функция добавления точки

Задание 5.11

Постройте функцию `AddPoint[BL,P]`, которая строит новую замкнутую ломаную путем добавления вновь поступившей точки-вершины `P` к заданной ломаной линии `BL`.

Выполнение задания 5.11

Согласно условию задачи, точки поступают в замкнутую ломаную линию последовательно одна за другой. Определим правило `BreakLine`, которое проводит инициализацию объекта:

```
BreakLine[] = BreakLine[0, {}, 0, 0];
```

Построим функцию `AddPoint`, добавляющую указанную точку в ломаную `BreakLine`. Функция имеет два аргумента: старая замкнутая ломаная и точка, которую нужно добавить. Результат работы функции – новая замкнутая ломаная с добавленной точкой. При добавлении точки изменяются свойства объекта: увеличивается количество точек, изменяется периметр и диаметр.

Согласно постановке задачи, точки поступают последовательно одна за другой, поэтому функция должна работать:

- с объектом, не содержащим точек;
- объектом, содержащим одну точку;

- объектом, содержащим много (более одной) точек.

Каждый из этих случаев обрабатываем отдельно. Для описания входных аргументов используем образцы (patterns).

Используем механизм верхних значений UpValues для закрепления вводимых правил за символом BreakLine, головой объекта ломаная линия:

BreakLine/:lhs:=rhs

Функцию AddPoint определим, задавая правила преобразований:

```
BreakLine/:AddPoint[
  (*объект ломаная, не имеющая вершин*),
  (*поступающая точка*)]:= "Add First Point";
BreakLine/:AddPoint[
  (*объект ломаная, содержащая одну вершину*),
  (*поступающая точка*)]:= "Add Second Point";
BreakLine/:AddPoint[
  (*объект ломаная, содержащая много вершин*),
  (*поступающая точка*)]:= "Add Next Point";
```

При описании входных параметров именуем не только весь образец, но и его части, которые будут использоваться в теле функции посредством имени.

Иницилируем объект замкнутая ломаная BL0:

BL0=BreakLine[]

Определим правило добавления первой точки:

```
BreakLine/:AddPoint[BreakLine[0, {}, rest__],
  P:{_, _}]:= BreakLine[1, {P}, rest];
BL1=AddPoint[BL0, {7, 7}]
BreakLine[1, {{7, 7}}, 0, 0]
```

Определим правило добавления второй точки в ломаную, содержащую одну точку. Обратим внимание на случай совпадения новой точки с уже существующей:

BreakLine /:

```
AddPoint[BL : BreakLine[1, {A : {_, _}}, rest___],  
  P : {_, _}] := Which[A == P, BL, True,  
  BreakLine[2, {A, P}, Distance[A, P] // N,  
    Distance[A, P]]]  
BL2=AddPoint[BL1,{20,9}]  
BreakLine[2,{{7,7},{20,9}},13.1529, $\sqrt{173}$  ]
```

Определим правило добавления новой точки в ломаную, содержащую две и более точки. Функция PointOnline, построенная в задании 5.9, поможет избежать случая, когда звенья, имеющие общую вершину, лежат на одной прямой.

```
BreakLine /: AddPoint[BL_BreakLine, P : {_, _}] :=  
  Which[PointOnline[BL[[2, {-2, -1}]], P], BL, True,  
    BreakLine[BL[[1]] + 1, Append[BL[[2]], P],  
      N[BL[[3]] + Distance[Last[BL[[2]]], P]],  
      Diameter[Append[BL[[2]], P]]]]
```

Тестируем работу функции AddPoint.

```
BL3 = AddPoint[BL2, {11, 14}]  
BreakLine[3, {{7, 7}, {20, 9}, {11, 14}},  
  23.4486, Diameter[{{7, 7}, {20, 9}, {11, 14}}]]  
BL4 = AddPoint[BL3, {1, 2}]  
BreakLine[4, {{7, 7}, {20, 9}, {11, 14}, {1, 2}},  
  39.0691, Diameter[{{7, 7}, {20, 9}, {11, 14}, {1, 2}}]]
```

Отообразим (рис. 5.13) ломаные линии BL3 и BL4, используем функцию Show, построенную в задании 5.7.

```
Show[#] & /@ {BL3, BL4}
```

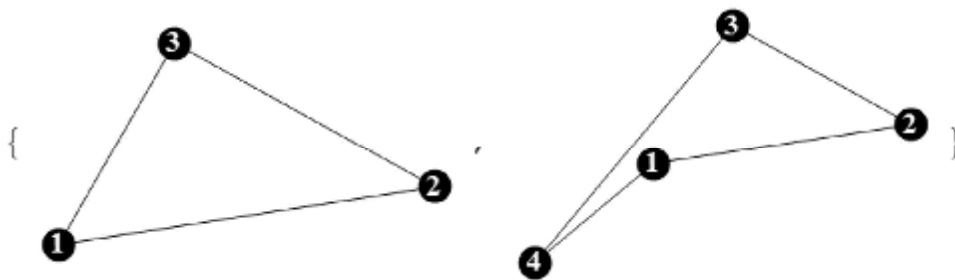


Рис. 5.3. Добавление четвертой точки к ломаной из трех точек

5.14. Анимация

Задание 5.12

Создайте мультфильм, демонстрирующий построение замкнутой ломаной последовательно поступающих точек плоскости. В кадре, помимо графического образа объекта, отобразите его свойства: количество вершин, их координаты, периметр замкнутой ломаной, диаметр множества ее угловых точек.

КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Режимы выполнения функций графики.
- 2) Функция Show.
- 3) Базовые функции двумерной графики.
- 4) Базовые функции трехмерной графики.
- 5) Двумерные графические примитивы.
- 6) Трехмерные графические примитивы.
- 7) Директивы функций графики.

ОГЛАВЛЕНИЕ

Лабораторная работа 5 (часть 3)	2
Задание 5 (часть 3)	2
5.11. Виртуальная модель замкнутой ломаной	2
Задание 5.9	2
Выполнение задания 5.9	2
5.12. Графическое представление замкнутой ломаной	4
Задание 5.10	4
Выполнение задания 5.10	4
5.13. Функция добавления точки	8
Задание 5.11	8
Выполнение задания 5.11	8
5.14. Анимация	11
Задание 5.12	11
Контрольные вопросы	11