

КРАТКО РЪКОВОДСТВО ЗА СИСТЕМАТА ЗА КОМПЮТЪРНА АЛГЕБРА WOLFRAM MATHEMATICA

Тихомир Иванов,

Институт по математика и информатика, БАН

Резюме. Настоящата статия е първа в поредицата „Математически модели на реални процеси и приложения на системите за компютърна алгебра за тяхното изследване“ и има по-скоро уводен характер. Основната ни цел е да представи една от най-популярните системи за компютърна алгебра – Wolfram Mathematica. Показани са някои базови възможности на системата, които биха били полезни в средния курс на образование, а запознаването с тях ще подготви читателя за последващите статии, свързани с математическото моделиране. Включени са и някои прости примери за самостоятелна работа, които позволяват усвояването на първоначални умения за работа със системата.

Keywords: mathematical modeling, computer algebra systems, education, Wolfram Mathematica

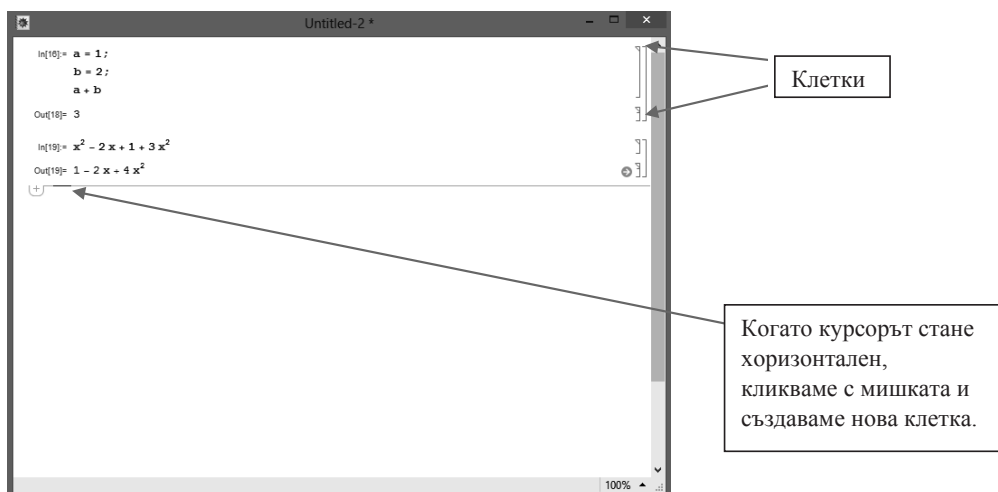
1. Основи на работата с Mathematica

Системата за компютърна алгебра Wolfram Mathematica е едно мощно средство, което позволява решаването на огромен набор от математически задачи, включително аритметични и алгебрични преобразования, решаване на алгебрични и диференциални уравнения и системи – точно и приближено, диференциране, интегриране и мн. др. Също така, Mathematica е удобно средство за визуализация, като предлага голям набор от възможности за построяването на двумерни и тримерни графики и за анимации. Системата позволява и написването на програми, които да извършват различни дейности. В настоящата работа ние ще се спрем на някои съвсем базови възможности (без да претендираме за каквато и да е изчерпателност) на системата Mathematica, които биха представлявали интерес за обучението по математика в средния курс на образование. Работата с тях може да бъде лесно овладяна, което прави системата подходяща за използване в този етап на обучение.

За да започнем работа със системата, ни е необходимо да създадем нов “notebook”. Това става от менюто *File -> New -> Notebook*. Работата в Mathematica е организирана в т.нар. клетки (cells). Всяка клетка съдържа команди, реализиращи определени функционалности и е оградена от квадратна скоба, намираща се в десния край на прозореца (Фиг. 1).

Създаването на нова клетка става, като преместим курсора под някоя от вече създадените, докато той стане хоризонтален, след което кликваме с левия бутон на мишката. Когато започнем да въвеждаме команди, на това място се появява нова клетка.

Изпълняването на командите в една клетка става, когато курсорът се намира в нея, с клавишната комбинация *Shift+Enter*. След като оценим съдържанието ѝ, пред нея се появява текст *In*, което означава, че това е клетка с инструкции. Клетките, в които се съдържат резултатите, пресметнати от Mathematica, се именуват с *Out*.



Фиг. 1

Нека сега се спрем на въпроса как се въвеждат алгебрични изрази в Mathematica. Можем да въвеждаме числа и букви (означаващи променливи и константи), свързани с аритметичните операции $+$, $-$, $*$, $/$, съответно събиране, изваждане, умножение, деление. Знакът за умножение може да се изпуска в редица случаи, когато това не предизвиква двусмислие (не можем да напишем например xu , защото системата ще възприеме това като един символ с име xu , но можем да напишем например $2x$). Знакът за степенуване е $^$, например x^2 . Можем да използваме и клавишната комбинация *Ctrl+^*, така че курсорът „да се вдигне“ и да получим израза x^2 . Квадратен корен може да се въведе като *Sqrt[израз]* или чрез клавишната комбинация *Ctrl+2*. При въвеждането на алгебрични изрази можем да използваме и вградените константи *Pi*, *E*, както и да указваме приоритета на операциите със скоби (задължително кръгли). Разгледайте следните примери:

```
In[1]:= x + 2 x - 4
      x * y - 12 / 7.4
      2 (x - 3) + 3 (x (y - 4) + 7)^2
      x^2 + Sqrt[x]
      x^2 + √x
```

Можем да използваме и редица вградени функции като *Sin*, *Cos*, *Tan*, *Cot*, *Log* и др. Имената на първите четири са достатъчно информативни. Ще отбележим само, че *Log* е вградена функция за намиране на натурален логаритъм. Обърнете внимание, че **имената на всички вградени функции се пишат с главна буква, а аргументите се ограждат в квадратни скоби**. Разгледайте следните примери и резултатите от оценяването на съответните изрази:

```
In[3]:= Sin[Pi] + Cos[Pi]
      Log[E]
      Tan[x] Cot[x]
```

```
Out[3]= -1
```

```
Out[4]= 1
```

```
Out[5]= 1
```

Можем и да запазваме данни в паметта по следния начин:

```
In[6]:= x = 2;
      y = 4;
```

След като се изпълни съдържанието на клетката с *Shift+Enter*, променливите *x* и *y* ще имат стойности съответно 2 и 4. Можем да ги използваме по-нататък в работата си:

```
In[8]:= x + y
```

```
Out[8]= 6
```

Да отбележим, че често е полезно да именуваме променливи с използването на долни индекси. Това може да стане, като използваме клавишната комбинация *Ctrl+_*, например

```
In[9]:= n0 = 1;
      n1 = 4;
      n2 = 2.5;
```

Обърнете внимание, че, ако веднъж сме присвоили стойност на дадена променлива, тя се пази в паметта, докато не я изтрием (или не излезем от програмата).

Нека в променливата x се пази стойност 2. Ако искаме да работим с x като със символ (например да пресметнем $\sin^2 x + \cos^2 x$ за произволно x), трябва да изтрием променливата от паметта. Това става по следния начин:

```
In[7]:= Clear[x]
```

Задача: Изпробвайте всички приведени примери.

Задача: Пресметнете $\ln 2981$ и $\cos \frac{319\pi}{13}$.

Вероятно резултатът от последната задача ви е изненадал. Тук е моментът да споменем, че Mathematica смята точно, докато не ѝ кажем друго. Числата се извеждат чрез точното си алгебрично представяне. Ако искаме да намерим приближение на дадено число във вид на десетична дроб, използваме вградената функция $N[\text{израз}]$, например:

```
In[33]:= N[Log[2981]]
```

```
N[Cos[319 Pi / 13]]
```

```
Out[33]= 8.00001
```

```
Out[34]= -0.120537
```

Полезно е да отбележим и че системата Mathematica има изключително пълна и удобна за работа документация, която може да бъде отворена от менюто *Help -> Documentation Center*. Винаги, когато имаме нужда да разберем как става нещо в Mathematica, можем да се обърнем към нея.

В документацията може да се търси по ключови думи, можем да напишем дадена команда, за да видим нейния синтаксис, можем и да се запознаем с възможностите на системата, благодарение на структурираните ръководства, които са подготвени за целта.

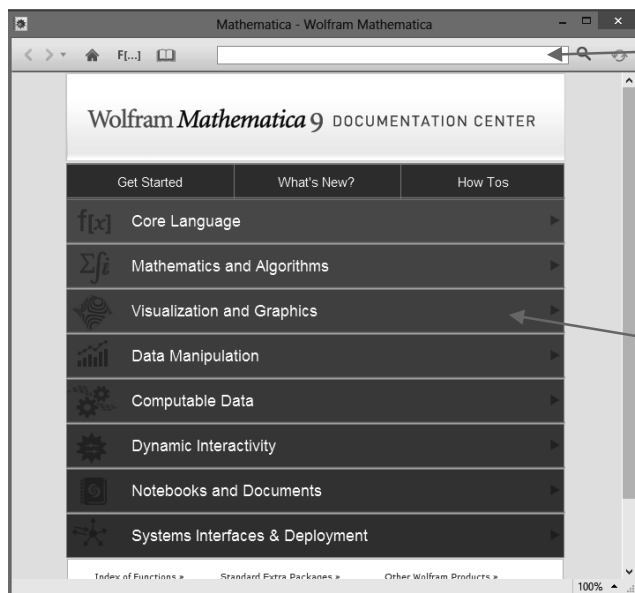
2. Работа с алгебрични изрази

Системата Mathematica е удобно средство, когато се налага да се извършват различни технически преобразования с алгебрични изрази, включително ирационални, тригонометрични и др. Ще разгледаме няколко основни вградени функции:

- $\text{Simplify}[\text{expr}]$ – опростява израза expr в съответствие с вградени в системата правила. Тук трябва да се отбележи, че невинаги идеята на Mathematica за „опростяване“ съвпада с нашата, но обикновено това е първата вградена функция, която се използва за целта.

```
In[14]:= Simplify[x (3 - x) - 5 x^2 + (x - 1) (2 x + 3)]
```

```
Out[14]= -3 + 4 x - 4 x^2
```



Тук можем да въведем ключови думи, по които да търсим в документацията.

В документацията могат да се открият и много добре структурирани ръководства, които да ни въведат в работата със системата и да ни запознаят с богатите ѝ възможности.

In[15]:= **Simplify** [**Cos** [2 x] + **Sin** [x]²]

Out[15]= **Cos** [x]²

• *Expand[expr]* – най-общо казано, тази вградена функция разкрива скобите в израза *expr*:

In[16]:= **Expand** [(x + 5)³ * (2 x - 1)²]

Out[16]= 125 - 425 x + 215 x² + 241 x³ + 56 x⁴ + 4 x⁵

• *Factor[poly]* – разлага полинома *poly* на прости множители:

In[17]:= **Factor** [125 - 425 x + 215 x² + 241 x³ + 56 x⁴ + 4 x⁵]

Out[17]= (5 + x)³ (-1 + 2 x)²

Задача: Опростете изразите $\ln(2e^5)$, $1 + \cos 2x$, $\frac{x + [x(x-1)]^3 - 4}{x^2 + x - 6}$.

3. Списъци. Автоматично генериране на списъци.

Основна структура данни в Mathematica е списъкът. Най-общо казано, той представлява последователност от елементи, разделени със запетаи и оградени от фигурни скоби, например {2, 3, 1, 4}, {x, 1, 1.5}. Както и всички други данни, списъците могат да се запишат в променлива:

```
In[18]:= list = {2, 3, 1, 4}
```

Ако искаме да използваме някои от елементите на списъка, това става по следния начин:

```
In[19]:= list[[1]]
          list[[3]] + list[[2]]
```

```
Out[19]= 2
```

```
Out[20]= 4
```

В горния пример изведохме първия елемент от списъка *list*, както и сумата на третия и втория елемент.

В много случаи е удобно, вместо да въвеждаме елементите на даден списък един по един, те да се създават автоматично по определено правило. Една възможност за това е вградената функция *Table*. Ще илюстрираме действието ѝ с няколко примера:

```
In[21]:= Table[k2, {k, 0, 10, 1}]
```

```
Out[21]= {0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
```

И така, първият аргумент на функцията *Table* е формулата, по която се изчисляват елементите на списъка. В случая тя зависи от променливата *k*. Вторият аргумент е т.нар. итератор, който указва как се изменя променливата – в случая *k* се мени от 0 до 10 през 1. С други думи, в горния пример създадохме списък с квадратите на числата от 0 до 10. Да разгледаме още един пример. Ще създадем списък, чиито елементи са от вида $\cos x - 0.1$, където *x* приема стойности от 0 до 2π през $\frac{\pi}{4}$:

```
In[22]:= Table[Cos[x] - 0.1, {x, 0, 2 Pi, Pi / 4}]
```

```
Out[22]= {0.9, 0.607107, -0.1, -0.807107, -1.1, -0.807107, -0.1, 0.607107, 0.9}
```

Последният пример, който ще разгледаме има за цел да ни покаже как можем да създадем списък от променливи, чиито стойности се пазят в паметта. Нека в паметта имаме променливите n_0, n_1, n_2, n_3 . Ще създадем списък от техните квадрати:

```
In[23]:= n0 = 2;
          n1 = 5;
          n2 = 7;
          n3 = 9;
          Table[nk2, {k, 0, 3, 1}]
```

```
Out[27]= {4, 25, 49, 81}
```

Тук правилото, по което се определят елементите на списъка, зависи от индекса *k*, който последователно приема стойности 0, 1, 2, 3.

Задача: Създайте списък от първите 50 нечетни числа.

Задача: Създайте списък от първите 20 прости числа. За целта използвайте вградената функция *Prime* – вижте синтаксиса и семантиката ѝ в документацията. Намерете сумата на квадратите на третото, седмото и петнадесетото от тях.

4. Решаване на уравнения и системи уравнения с Mathematica

В Mathematica можем да решаваме уравнения с помощта на функцията *Solve*. Тя има следния синтаксис: *Solve[equation, var]*, където *equation* е уравнението, което искаме да решим, а *var* – неизвестното, спрямо което решаваме уравнението. Обърнете внимание, че, когато въвеждаме уравнението, използваме `==` вместо `=`. Системата Mathematica се справя с много класове уравнения, които могат да бъдат решени точно, включително такива, които имат комплексни корени. Нека разгледаме следните примери:

```
In[28]:= Solve[6 x^3 - 23 x^2 + 25 x - 6 == 0, x]
```

```
Out[28]:= {{x -> 1/3}, {x -> 3/2}, {x -> 2}}
```

```
In[29]:= Solve[Sqrt[x - 1] + x == 4 + Sqrt[x + 4], x]
```

```
Out[29]:= {{x -> 5}}
```

```
In[30]:= Solve[x^4 - 2 x^3 - x^2 + 6 x - 6 == 0, x]
```

```
Out[30]:= {{x -> 1 - I}, {x -> 1 + I}, {x -> -Sqrt[3]}, {x -> Sqrt[3]}}
```

В някои случаи, когато корените на уравнението не могат да бъдат представени точно или представянето им е прекалено сложно (и съответно ни носи малко информация), можем да използваме функцията *NSolve*. Тя има същия синтаксис, но връща приближения на корените във вид на десетични дробни. Сравнете следните примери:

```
In[31]:= Solve[x^5 - 10 x^2 + 5 x + 1 == 0, x]
```

```
Out[31]:= {{x -> Root[1 + 5 #1 - 10 #1^2 + #1^5 &, 1]},
           {x -> Root[1 + 5 #1 - 10 #1^2 + #1^5 &, 2]}, {x -> Root[1 + 5 #1 - 10 #1^2 + #1^5 &, 3]},
           {x -> Root[1 + 5 #1 - 10 #1^2 + #1^5 &, 4]}, {x -> Root[1 + 5 #1 - 10 #1^2 + #1^5 &, 5]}}
```

```
In[32]:= NSolve[x^5 - 10 x^2 + 5 x + 1 == 0, x]
```

```
Out[32]:= {{x -> -1.22065 - 1.89169 I}, {x -> -1.22065 + 1.89169 I},
           {x -> -0.153102}, {x -> 0.66946}, {x -> 1.92494}}
```

In[33]:= **Solve** [$x^3 - 10x^2 + 5x + 1 == 0$, x]

Out[33]= $\left\{ \left\{ x \rightarrow \frac{1}{3} \left(10 + \frac{85}{\left(\frac{1}{2} (1523 + 9i\sqrt{1691}) \right)^{1/3}} + \left(\frac{1}{2} (1523 + 9i\sqrt{1691}) \right)^{1/3} \right) \right\}, \right.$
 $\left\{ x \rightarrow \frac{10}{3} - \frac{1}{6} (1 + i\sqrt{3}) \left(\frac{1}{2} (1523 + 9i\sqrt{1691}) \right)^{1/3} - \frac{85 (1 - i\sqrt{3})}{3 \times 2^{2/3} (1523 + 9i\sqrt{1691})^{1/3}} \right\},$
 $\left. \left\{ x \rightarrow \frac{10}{3} - \frac{1}{6} (1 - i\sqrt{3}) \left(\frac{1}{2} (1523 + 9i\sqrt{1691}) \right)^{1/3} - \frac{85 (1 + i\sqrt{3})}{3 \times 2^{2/3} (1523 + 9i\sqrt{1691})^{1/3}} \right\} \right\}$

In[34]:= **NSolve** [$x^3 - 10x^2 + 5x + 1 == 0$, x]

Out[34]= $\{ \{x \rightarrow -0.152671\}, \{x \rightarrow 0.692369\}, \{x \rightarrow 9.4603\} \}$

По същия начин можем да решаваме и системи уравнения. Тогава първият аргумент на **Solve**/**NSolve** е списък с уравнения, разделени със запетаи и оградени с фигурни скоби $\{ \}$, а вторият – списък с неизвестните, относно които решаваме:

In[38]:= **Solve** [$\{x^2 - y == 1, -x + y == 1\}$, $\{x, y\}$]

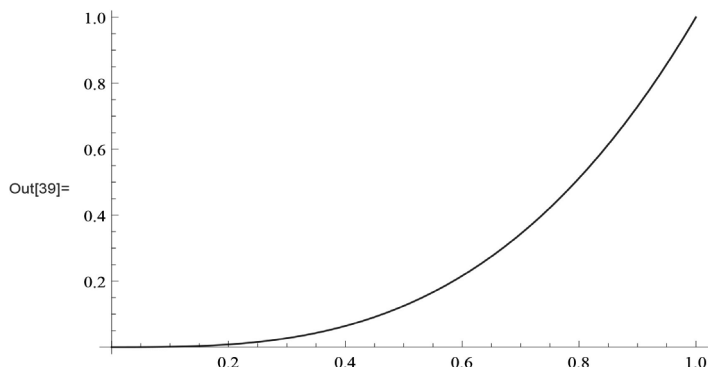
Out[38]= $\{ \{x \rightarrow -1, y \rightarrow 0\}, \{x \rightarrow 2, y \rightarrow 3\} \}$

5. Графики в Mathematica

Да видим сега как с помощта на системата Mathematica можем да построяваме графики на функции. Командата за това е *Plot*. Тя приема два аргумента – първият е функцията (или списък от функции), чиято графика искаме да построим. Вторият е списъкът $\{var, from, to\}$, където *var* е името на независимата променлива във функцията, а $[from, to]$ е интервалът, в който искаме да бъде построена графиката.

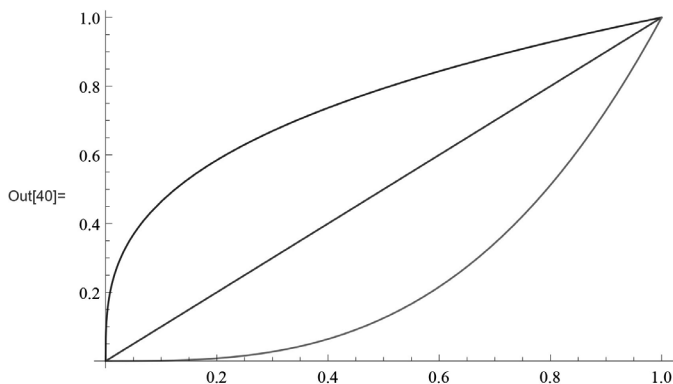
Например, ако искаме да построим графиката на функцията $f(x) = x^3$, това става по следния начин:

In[39]:= **Plot** [x^3 , $\{x, 0, 1\}$]



Ако искаме да построим графиките на повече от една функция, това става, като за първия аргумент на *Plot* подадем списък от функции:

```
In[40]:= Plot[{x1/3, x, x3}, {x, 0, 1}]
```

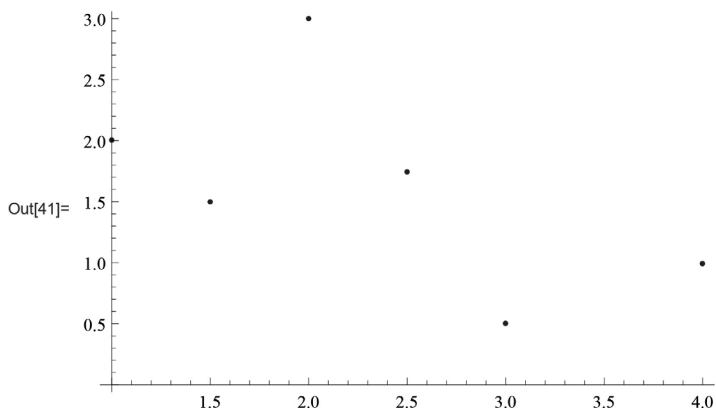


Задача: Изследвайте графично броя на различните корени на уравнението $|x^2 - 4x + 1| = a$ в зависимост от стойностите на параметъра a . (Вградената функция за намиране на абсолютна стойност е *Abs*)

Задача: Решете графично уравнението $x^2 - 4x + 1 = \sqrt{x + 3}$. Намерете приблизително корените на това уравнение. За целта визуализирайте графиките в достатъчно малък интервал, съдържащ съответния корен, за да можете да определите с добра точност абсцисата на пресечната точка на графиките на лявата и дясната страна.

Когато искаме да изобразим множество от точки в равнината, използваме вградената функция *ListPlot*, която приема един аргумент – списък с наредени двойки от вида $\{x, y\}$ – координатите на точките:

```
In[41]:= ListPlot[{{1, 2}, {1.5, 1.5}, {2, 3}, {2.5, 1.75}, {3, 0.5}, {4, 1}}]
```



Често е удобно списъкът с точки да се запише в една променлива и тя да се подаде като аргумент на функцията *ListPlot*. Да разгледаме следния пример. Нека в паметта имаме следните променливи:

```
In[42]:= n0 = 1;
        n1 = 4;
        n2 = 2.5;
```

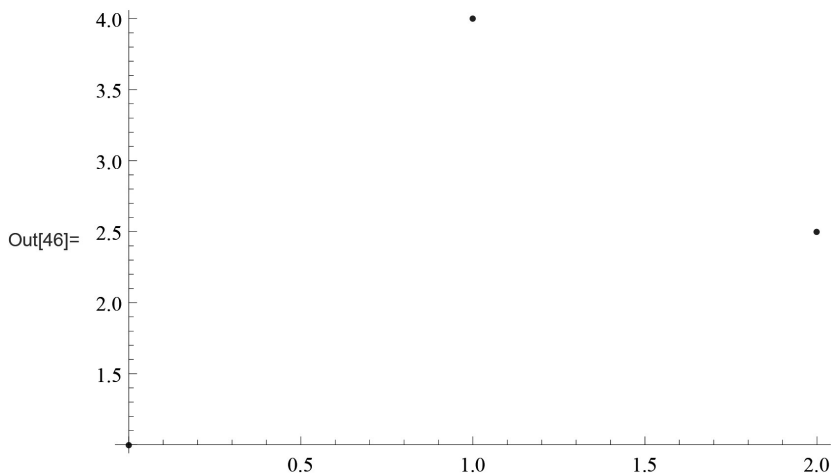
Първо ще създадем списък с наредени двойки от вида $\{k, n_k\}$, където k последователно приема стойности 0, 1, 2 и ще го запазим в променлива, която ще наречем *pts*:

```
In[45]:= pts = Table[{k, nk}, {k, 0, 2, 1}]
```

```
Out[45]= {{0, 1}, {1, 4}, {2, 2.5}}
```

Накрая ще използваме така дефинирания списък и ще изобразим точките от него:

```
In[46]:= ListPlot[pts]
```



6. Построяване на итерационен процес в Mathematica

В този параграф ще демонстрираме как може да се използва операторът за цикъл *Do*. Ще направим това на базата на един пример. Ще покажем как можем да намерим първите 15 члена от редицата на Фибоначи:

```
In[47]:= n0 = 1;
        n1 = 1;
        Do[
            nk = nk-1 + nk-2,
            {k, 2, 15}
        ]
```

Първо, запазваме в променливите n_0 и n_1 стойностите на първите два елемента от редицата на Фибоначи. След това последователно намираме членовете n_k , където k приема последователно стойности от 2 до 15. Считаме, че примерът е достатъчно показателен и затова няма да се спираме по-подробно върху синтаксиса на оператора *Do*. Ще отбележим само, че ако искаме на всяка итерация да се извършва повече от една операция, това става, като отделните операции се разделят с точка и запетая, а след последната (преди итератора) се постави запетая.

Задача: С помощта на командата *ListPlot* изобразете първите 15 члена от редицата на Фибоначи.

Задача: Намерете първите 20 члена от геометрична прогресия с първи член $b_0 = 2$ и частно $q = 0.9$, като използвате формулата за общия i член. Илюстрирайте графично.

7. Заключение

В настоящата статия се запознахме с работата със системата за компютърна алгебра *Wolfram Mathematica* и разгледахме няколко основни вградени функции. Това е една добра основа, която може да се използва при решаването на различни задачи от средния курс на обучение по математика. Разбира се, трябва да отбележим, че дори за функциите, които разгледахме, пропуснахме редица особености, които биха били важни в една или друга ситуация. В следващите статии от поредицата „Математически модели и приложения на системите за компютърна алгебра за тяхното изследване“ ще коментираме някои аспекти от работата със системата в конкретен контекст. Освен това, надяваме се, че читателят ще продължи да се усъвършенства в използването на *Mathematica* с помощта на много богатата ѝ документация, както и съществуващата литература.

ЛИТЕРАТУРА

1. S. Hollis (2008). *Manual for Stewart's Single Variable Calculus*. Brooks/Cole.
2. S. Wagon (2010). *Mathematica in Action. Problem Solving Through Visualization and Computation*. Springer.

REFERENCES:

1. S. Hollis (2008). *Manual for Stewart's Single Variable Calculus*. Brooks/Cole.
2. S. Wagon (2010). *Mathematica in Action. Problem Solving Through Visualization and Computation*. Springer.

A SHORT MANUAL FOR THE COMPUTER ALGEBRA SYSTEM WOLFRAM MATHEMATICA

Abstract. This is the first (introductory) article in the series “Mathematical models of real processes and applications of computer algebra systems to their study”. We present one of the most popular computer algebra systems—Wolfram Mathematica. Some of its basic capabilities that could be useful for the high school education in mathematics are shown. Studying the latter is a necessary background that will prepare the reader for the following articles, dealing with mathematical modeling. Also, a few simple problems are included to develop some basic skills for working with the system.

✉ **Mr. Tihomir Ivanov**

Institute of Mathematics and Informatics

Bulgarian Academy of Sciences

Sofia, Bulgaria

E-mail: tihomir_ivanov@ymail.com