

pher Modification. International Journal of Information Assurance and Security (JIAS), V.15, N4, 2020, pp.165-174.

References

1. C. F. Van Loan. The ubiquitous Kronecker product. Journal of Computational and Applied Mathematics, 123, 2000, p. 85–100.
2. C. Koukouvinos, E. Lappas, D. E. Simos. Encryption schemes using orthogonal arrays, Journal of Discrete Mathematical Sciences & Cryptography, 12 (5), 2009, pp. 615–628.
3. Alexander Chefranov, Evgeny Dukhnich. One-Time Kronecker Product-Based Hill Cipher Modification.

- International Journal of Information Assurance and Security (JIAS), V.12, N3, 2017, pp.94-103.
4. Dukhnich E.I., Shapel A.P. An effective method for the formation of involutive matrices for the modified Hill cipher, Materials of the National Conference "Scientific, technical, economic and legal aspects of the development of the transport complex", 2019, part 1, pp. 71-72
5. Alexander Chefranov, Evgeny Dukhnich, Alexander Shapel. One-Time Involutive Matrix-Based Hill Cipher Modification. International Journal of Information Assurance and Security (JIAS), V.15, N4, 2020, pp.165-174.

УДК 53.05; 004.02

DOI: 10.34046/aumsuomt95/24

ВИЗУАЛИЗАЦИЯ СЛОЖЕНИЯ КОЛЕБАТЕЛЬНЫХ ДВИЖЕНИЙ МЕТОДОМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Е.Н. Сюсюка, кандидат технических наук, доцент

А. А. Колесников, курсант

А.Е. Чупин, курсант

В статье представлены методы сложения гармонических колебаний в Pascal ABC и Microsoft Excel с оживлением и возможностью изменения по алгоритму. Цель данной работы заключается в ознакомлении читателей о технологии сложения гармонических колебаний, как направленных вдоль одной прямой, так и взаимно перпендикулярных, а также реализации этих технологий средствами Pascal ABC и Microsoft Excel. В статье приведены результаты исследований, которые дают представления о реализации сложения гармонических колебаний средствами Pascal ABC и Microsoft Excel. Данный алгоритм можно использовать для моделирования наложения волновых процессов как в любой упругой среде при изучении устойчивости судна на воде, конструкций, вибродиагностики механизмов, так и для анализа интерференционных явлений в оптике и при распространении электромагнитных волн радио- и оптической связи. Кроме того, методику можно применять с целью визуализации моделирования сложения колебаний в учебном процессе.

Ключевые слова: гармонические колебания, сложение гармонических колебаний, направленные вдоль одной прямой колебания, взаимно перпендикулярные колебания, фигуры Лиссажу, Pascal ABC, Microsoft Excel.

The article presents methods of addition of harmonic oscillations in Pascal ABC and Microsoft Excel with revival and possibility of change by algorithm. The purpose of this work is to familiarize readers about the technologies of addition of harmonic oscillations, both directed along one straight line and mutually perpendicular, as well as the implementation of these technologies by means of Pascal ABC and Microsoft Excel. The article presents the results of studies that give an idea of the implementation of the addition of harmonic oscillations by means of Pascal ABC and Microsoft Excel.

Keywords: harmonic oscillations, addition of harmonic oscillations, directed along one straight line oscillations, mutually perpendicular oscillations, lissajou figures, Pascal ABC, Microsoft Excel.

Актуальность визуализации сложения колебаний методом информационных технологий

Различные виды колебаний имеют совершенно разную физическую природу и объединяются лишь единством их математического описания. Естественно, что при таком широком взгляде на проблему не удастся в рамках одного учебного предмета отразить специфику колебательных явлений в каждом из рассматриваемых областей техники. Изучение колебательных процессов имеет большое значение для развития современной техники, так как с её помощью могут быть корректно рассмотрены практически важные проблемы создания систем стабилизации,

измерения вибрационных характеристик и т.п. Это позволяет проектировать приборы и системы, способные функционировать на подвижных объектах, находя свое применение в авиации, судостроении и других областях техники.

На море, как известно, вода всегда находится в движении. В основном всегда, мы видим волны на поверхности воды. Колебания ей придают подземные толчки, движение воздушных масс, гравитационная сила Луны, течения, взаимные движения судов и т.д. Когда колебания в воде идут сонаправленно и с одинаковой частотой, происходит сложение волн, и высота волны увеличивается. Однако, если одной водной

массе сообщают равные и противоположно направленные колебания на поверхности в сторону берега, в придонной зоне – в сторону моря. Так, цикличность волнового процесса нарушается, создается гидравлическое сопротивление, что приводит к уменьшению амплитуды колебаний, что важно для прогнозирования волновых явлений при возведении пристаней, судоходстве, строительстве и т.д.

Теория сложения колебаний важна так же для вибродиагностики, которая в настоящее время является наиболее распространенным методом диагностики подшипников редукторов по уровню ударной вибрации и других механизмов.

Гармоническими колебаниями называются движения или процессы, которые характеризуются повторяемостью во времени, при которых колеблющаяся величина изменяется со временем по закону синуса или косинуса

$$S = A \sin(\omega t + \varphi_0)$$

$$S = A \cos(\omega t + \varphi_0)$$

где A – амплитуда - максимальное значение колеблющейся величины;

$(\omega t + \varphi_0)$ – фаза колебаний в момент времени t ;

φ_0 – начальная фаза колебаний в момент времени $t=0$, $[\varphi] = \text{рад}$;

ω – круговая (циклическая) частота – число полных колебаний, выраженных в радианной (угловой) мере, за 1 секунду.

$$\omega = \frac{2\pi}{T} = 2\pi\nu \quad [\omega] = \frac{\text{рад}}{\text{с}}$$

Период колебаний – промежуток времени, за который совершается одно полное колебание и фаза колебания получает приращение 2π :

$$T = \frac{2\pi}{\omega}, \quad [T] = \text{с}$$

Частота колебаний – число полных колебаний за 1 секунду (величина, обратная периоду колебаний):

$$\nu = \frac{1}{T}, \quad [\nu] = \frac{1}{\text{с}} = \text{с}^{-1} = \text{Гц}$$

Под сложением колебаний понимают нахождение закона результирующих колебаний системы в тех случаях, когда эта система одновременно участвует в нескольких колебательных процессах.

Различают два предельных случая: 1) сложение колебаний одинакового направления и одинаковой частоты; 2) сложение взаимно перпендикулярных колебаний.

Сложение двух одинаково направленных гармонических колебаний

Сложим гармонические колебания одного направления и одинаковой частоты

$$\begin{cases} x_1 = A_1 \cos(\omega_0 t + \varphi_1) \\ x_2 = A_2 \cos(\omega_0 t + \varphi_2) \end{cases}$$

воспользовавшись методом вращающегося вектора амплитуды. Построим векторные диаграммы этих колебаний. Так как векторы A_1 и A_2 вращаются с одинаковой угловой скоростью ω_0 , то разность фаз $(\varphi_2 - \varphi_1)$ между ними остается постоянной.

• Уравнение результирующего колебания имеет вид:

$$x = x_1 + x_2 = A \cos(\omega_0 t + \varphi)$$

• Амплитуда A результирующего колебания равна:

$$A^2 = A_1^2 + A_2^2 + 2A_1A_2 \cos(\varphi_2 - \varphi_1)$$

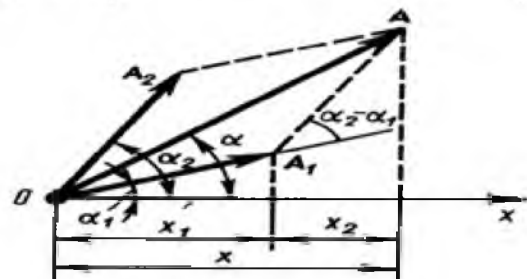


Рисунок 1 – Метод векторного сложения амплитуд

Амплитуда результирующего колебания зависит от разности фаз $(\varphi_2 - \varphi_1)$ складываемых колебаний:

а) если $(\varphi_2 - \varphi_1) = \pm 2m\pi$ ($m=0, 1, 2, \dots$), то амплитуда результирующего колебания равна сумме амплитуд складываемых колебаний:

$$A = A_1 + A_2;$$

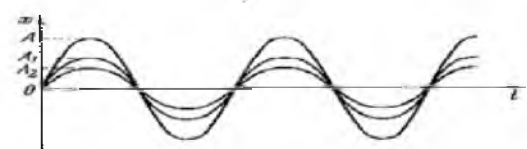


Рисунок 2 – Результат сложения колебаний, направленных вдоль одной прямой

б) если $(\varphi_2 - \varphi_1) = \pm (2m + 1)\pi$ ($m = 0, 1, 2, \dots$), то амплитуда результирующего колебания равна разности амплитуд складываемых колебаний:

$$A = |A_1 - A_2|$$

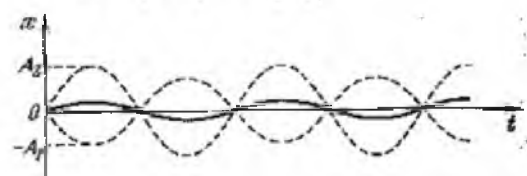


Рисунок 3

• Начальная фаза результирующего колебания:

$$\operatorname{tg} \varphi = \frac{A_1 \sin \varphi_1 + A_2 \sin \varphi_2}{A_1 \cos \varphi_1 + A_2 \cos \varphi_2}$$

Сложение взаимно перпендикулярных колебаний одинаковой частоты

Пусть точка М одновременно колеблется вдоль осей координат ОХ и ОУ по законам:

$$\begin{cases} x = A_1 \sin(\omega t + \varphi_1) \\ y = A_2 \sin(\omega t + \varphi_2) \end{cases}$$

где x и y – декартовы координаты точки М.

Уравнение траектории результирующего движения точки М в плоскости ХОУ можно найти, исключив из выражений для x и y параметр t:

$$\frac{x^2}{A_1^2} + \frac{y^2}{A_2^2} - \frac{2xy}{A_1 A_2} \cos(\varphi_2 - \varphi_1) = \sin^2(\varphi_2 - \varphi_1)$$

Траектория имеет форму эллипса, причем точка М описывает этот эллипс за время, равное периоду складываемых колебаний

$$T = \frac{2\pi}{\omega}$$

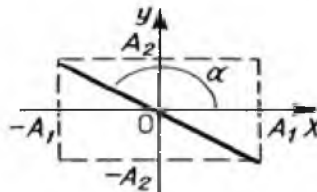
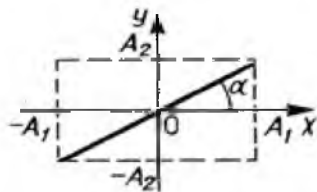


Рисунок 5

если $A_1 = A_2$, то траектория точки М представляет собой окружность, такое движение точки М называют циркулярно поляризованными или поляризованными по кругу.

б) если $(\varphi_2 - \varphi_1) = m\pi$ ($m = 0, \pm 1, \pm 2, \dots$), эллипс вырождается в отрезок прямой (рисунок 5):

$$y = \pm \frac{A_2}{A_1} x$$

Знак «+» соответствует четным значениям m , т.е. сложению синфазных колебаний, а знак «-» – нечетным значениям m , т.е. сложению колебаний, происходящих в противофазе.

В этих случаях точка М совершает линейно поляризованные колебания. Она колеблется с частотой складываемых колебаний и амплитудой:

$$A = \sqrt{A_1^2 + A_2^2}$$

вдоль прямой линии, составляющей с осью ОХ угол:

Поэтому результирующее движение точки М называют эллиптически поляризованными колебаниями.

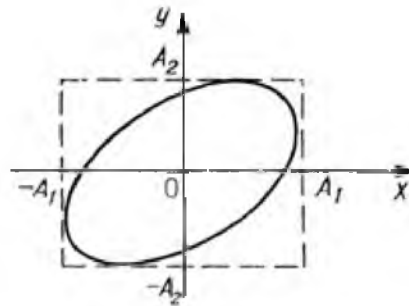


Рисунок 4

Ориентация осей эллипса (рисунок 4) и его размеры зависят от амплитуд складываемых колебаний A_1, A_2 и разности фаз $(\varphi_2 - \varphi_1)$:

а) если $(\varphi_2 - \varphi_1) = (2m+1)\frac{\pi}{2}$ ($m = 0, \pm 1, \pm 2, \dots$), то оси эллипса совпадают с осями координат ОХ и ОУ, а размеры его полуосей равны амплитудам A_1 и A_2 :

$$\frac{x^2}{A_1^2} + \frac{y^2}{A_2^2} = 1$$

$$\alpha = \operatorname{arctg} \left(\frac{A_2}{A_1} \cos m\pi \right)$$

Если частоты взаимно перпендикулярных колебаний не одинаковы и относятся как целые числа, то траектории результирующего движения имеют более сложные формы. Их называют фигурами Лиссажу.

На рисунке представлены фигуры Лиссажу для различных соотношений частот (указаны слева) и разностей фаз (указаны вверх); разность фаз принимается равной Φ .

Процесс распространения колебаний в сплошной среде периодический во времени и пространстве называется волновым процессом (или волной).

Сплошная среда – среда, непрерывно распределенная в пространстве и обладающая упругими свойствами.

Основным свойством всех волн, независимо от их природы, является перенос энергии без переноса вещества.

Упругие (или механические) волны – механические возмущения, распространяющиеся в упругой среде.

Упругая среда – вещество, способное подвергаться упругим деформациям.

Примеры упругих волн – звуковые волны, волны на воде. Упругие волны бывают продольные и поперечные.

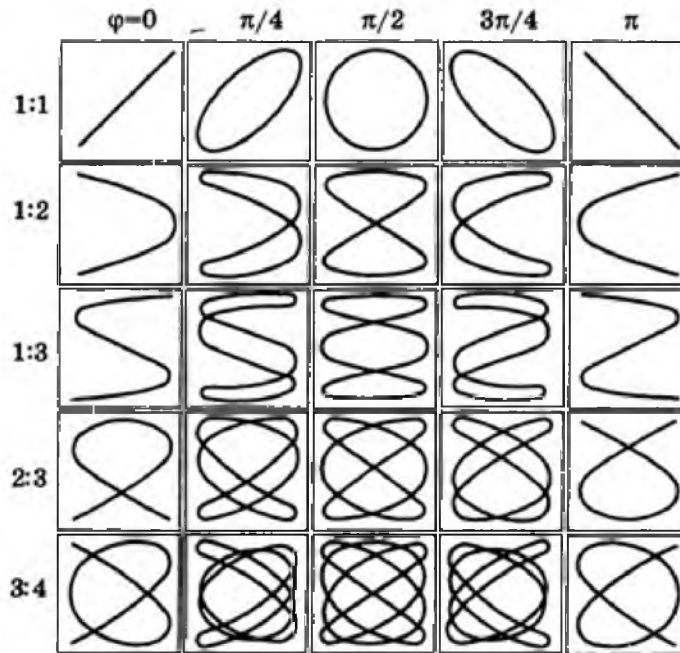


Рисунок 6 – Фигуры Лиссажу- результат взаимно перпендикулярных колебаний

Современные информационные технологии позволяют произвести моделирование этих явлений. Мы рассмотрим применение таких средств программирования, как PascalABC и Excel[2, 3, 1].

Сложение направленных вдоль одной прямой гармонических колебаний в PascalABC с оживлением и возможностью изменения по алгоритму.

Нам даны следующие величины: скорость прокрутки волны (s), амплитуды первой и второй волн (A_1, A_2), периоды колебаний первой и второй волн (T_1, T_2), начальные фазы первой и второй волн (φ_1, φ_2).

1. В программе PascalABC[7] загружаем нужные нам для работы модули и декларируем необходимые переменные:

```

program Waves;
uses VCL, Utils;
var
MainForm: Form;
y, maxw: integer;
SE1, SE2, SE3, SE4, SE5, SE6, SE7: SpinEdit;
CG1, CG2, CG3: ColorGrid;
BM: Bitmap;
CT: CanvasType;
PB: PaintBox;
T: Timer;
2. Записываем функции, которые упростят процесс добавления полей для ввода исходных числовых значений и для выбора цветов:
function addText(s: string): TLabel;
begin
Result:=TLabel.Create(0, y, s);
if Result.Width>maxw then maxw:=Result.Width;
Inc(y, Result.Height)
end;
function addSpinEdit(i: integer): SpinEdit;
begin

```

```

Result:=SpinEdit.Create(0, y, IntToStr(i));
ifResult.Width>maxw then maxw:=Result.Width;
Inc(y, Result.Height)
end;
functionaddColorGrid(i: integer): ColorGrid;
begin
Result:=ColorGrid.Create(0, y);
Result.BackgroundEnabled:=false;
Result.ForegroundIndex:=i;
ifResult.Width>maxw then maxw:=Result.Width;
Inc(y, Result.Height)
end;

```

3. Прописываем функцию, вычисляющую координату y по координате x с учетом амплитуды, периода, начальной фазы волны, скорости прокрутки пикселей и времени, прошедшего с начала запуска программы:

```

function f(A, T, fi, x, v: integer; s: real): integer;
begin
Result:=trunc(A*sin(2*pi*(x+v*s)/T+fi*pi/180))
end;

```

В математическом виде эта функция выглядела бы так:

$$f = \left[A \cdot \sin \left(\frac{2\pi(x+vs)}{T} + \frac{\varphi\pi}{180} \right) \right] \quad (1.1)$$

Где A – амплитуда волны в пикселях, T – период колебаний волны в пикселях, φ – начальная фаза волны, x – координата x , обозначающая время, в пикселях, v – скорость прокрутки волны в пикселях в секунду, s – количество секунд, прошедших с момента запуска программы, квадратные скобки – целая часть от числа.

4. Прописываем процедуру прорисовки графика с использованием двойной буферизации (чтобы не было т.н. "моргания")[5]:

```

procedureDraw;
var A1, T1, fi1, A2, T2, fi2, v, x, y: integer; color1, color2, color3: ColorType; s: real;
begin
A1:=SE1.Value; //считываем амплитуду первой волны из поля ввода
T1:=SE2.Value; //считываем период колебаний первой волны из поля ввода
fi1:=SE3.Value; //считываем начальную фазу первой волны из поля ввода
color1:=CG1.ForegroundColor; //считываем цвет первой волны из поля выбора цвета
//те же данные получаем о второй волне
A2:=SE4.Value;
T2:=SE5.Value;
fi2:=SE6.Value;
color2:=CG2.ForegroundColor;
//получаем цвет суммарной волны и скорость прокрутки
color3:=CG3.ForegroundColor;
v:=SE7.Value;
with CT do
begin
withBMdo
begin
Clear; //очищаем буфер, далее все графические функции будут реализовываться в буфере
s:=Milliseconds/1000; //считываем кол-во секунд с момента запуска программы (кол-во миллисекунд / 1000)
//рисует первую волну
Pen.Color:=color1; //устанавливаем цвет пера
forx:=0 to Width-1 do //проходим по циклу все координаты x, которые можно отобразить на экране
begin
y:=f(A1, T1, fi1, x, v, s); //вычисляем координату y по формуле

```

```
if x=0 then MoveTo(x, y+Height div 2) else LineTo(x, y+Height div 2) //проводим линию от предыдущей до следующей точки или перемещаем перо в начальную точку, если у нас x пока равен 0 (к координате у прибавляем половину высоты экрана, чтобы ось Ox проходила по середине экрана)
```

```
end;
```

```
    //рисует вторую волну по тому же принципу
```

```
Pen.Color:=color2;
```

```
for x:=0 to Width-1 do
```

```
begin
```

```
y:=f(A2, T2, fi2, x, v, s);
```

```
if x=0 then MoveTo(x, y+Height div 2) else LineTo(x, y+Height div 2)
```

```
end;
```

```
    //рисует суммарную волну по тому же принципу, только у получаем суммированием двух предыдущих волн
```

```
Pen.Color:=color3;
```

```
for x:=0 to Width-1 do
```

```
begin
```

```
y:=f(A1, T1, fi1, x, v, s)+f(A2, T2, fi2, x, v, s);
```

```
if x=0 then MoveTo(x, y+Height div 2) else LineTo(x, y+Height div 2)
```

```
end;
```

```
PB.Draw(0, 0, BM) //рисует буфер в графическом окне
```

```
end
```

```
end
```

```
end;
```

5. Создаём основную программу:

```
begin
```

```
MainForm:=Form.Create('Волны'); //Создаём главную форму
```

```
MainForm.Width:=1000; //устанавливаем ширину этой формы в пикселях
```

```
MainForm.Height:=600; //устанавливаем высоту этой формы в пикселях
```

```
y:=0;
```

```
maxw:=0;
```

```
    //добавляем в форму все необходимые поля
```

```
addText('Амплитуда первой волны (в пикселях):');
```

```
SE1:=addSpinEdit(200);
```

```
addText('Период первой волны (в пикселях):');
```

```
SE2:=addSpinEdit(200);
```

```
addText('Начальная фаза первой волны (в градусах):');
```

```
SE3:=addSpinEdit(0);
```

```
addText('Цветпервойволны:');
```

```
CG1:=addColorGrid(9);
```

```
addText('Амплитуда второй волны (в пикселях):');
```

```
SE4:=addSpinEdit(140);
```

```
addText('Период второй волны (в пикселях):');
```

```
SE5:=addSpinEdit(200);
```

```
addText('Начальная фаза второй волны (в градусах):');
```

```
SE6:=addSpinEdit(180);
```

```
addText('Цветвторойволны:');
```

```
CG2:=addColorGrid(12);
```

```
addText('Цветсуммарнойволны:');
```

```
CG3:=addColorGrid(13);
```

```
addText('Скорость прокрутки волн (в пикселях в секунду):');
```

```
SE7:=addSpinEdit(80);
```

```
BM:=Bitmap.Create(600, 400); //создаёмбуфер
```

```
CT:=BM.Canvas;
```

```
PB:=PaintBox.Create(maxw+50, 50, 600, 400); //создаёмграфическоеокноформе
```

T:=Timer.Create(Draw, 100) //запускаем таймер, который будет каждые 100 миллисекунд вызывать процедуру прорисовки графика в графическом окне
end.

6. Амплитуда результирующего колебания зависит от разности фаз ($\varphi_2 - \varphi_1$) складываемых колебаний:

а) если $(\varphi_2 - \varphi_1) = \pm(2m+1)\pi$ ($m = 0, 1, 2, \dots$), то амплитуда результирующего колебания равна разности амплитуд складываемых колебаний: $A = |A_1 - A_2|$ (1.2)

Запускаем программу и меняем некоторые параметры так, чтобы волны колебались в противофазе ($\varphi_2 - \varphi_1 = \pi$):

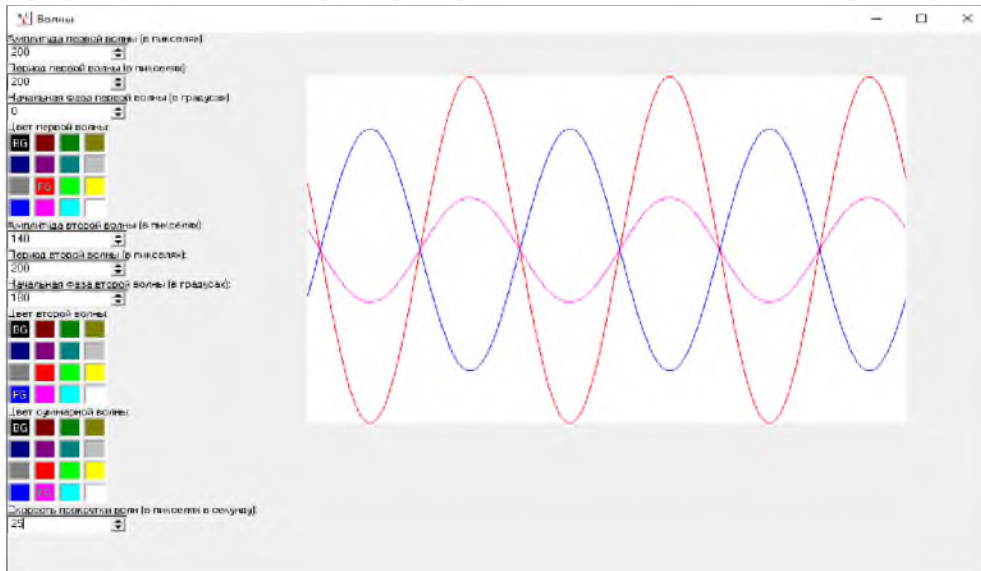


Рисунок 7 – Сложение колебаний в противофазе

б) если $(\varphi_2 - \varphi_1) = \pm 2m\pi$ ($m = 0, 1, 2, \dots$), то амплитуда результирующего колебания равна сумме амплитуд складываемых колебаний: $A = A_1 + A_2$ (1.3). Меняем параметры так, чтобы волны колебались в одной фазе ($\varphi_2 - \varphi_1 = 0$):

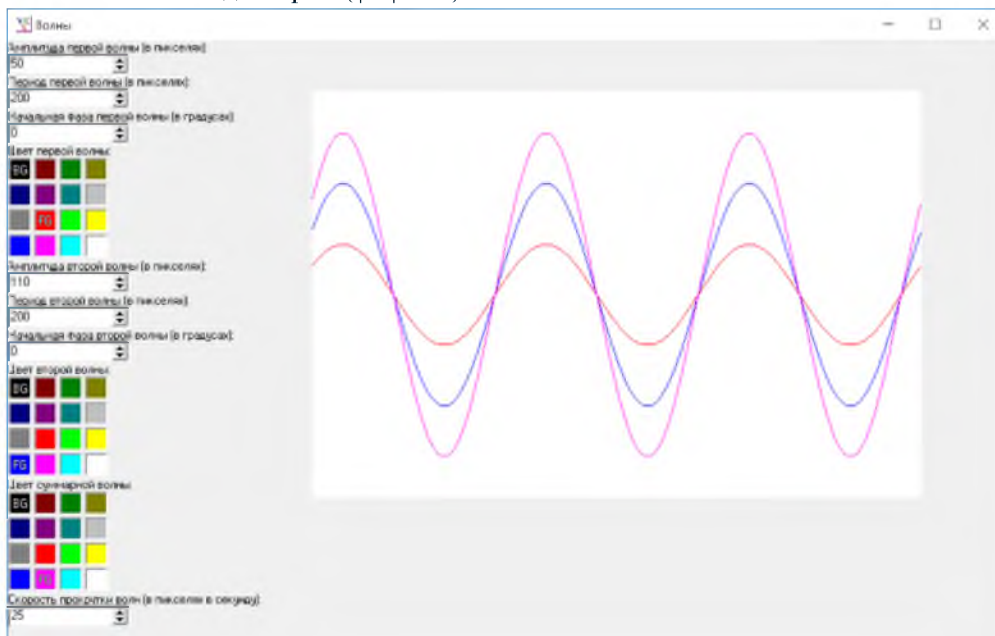


Рисунок 8 – Сложение колебаний в одной фазе

с) Меняем параметры так, чтобы волны колебались с разностью фаз $\frac{\pi}{4}$:

Амплитуда A результирующего колебания равна:

$$A = \sqrt{A_1^2 + A_2^2 + 2A_1A_2 \cos(\varphi_2 - \varphi_1)} = \sqrt{A_1^2 + A_2^2 + 2A_1A_2 \cos \frac{\pi}{4}} = \sqrt{A_1^2 + A_2^2 + A_1A_2\sqrt{2}} \quad (1.4)$$

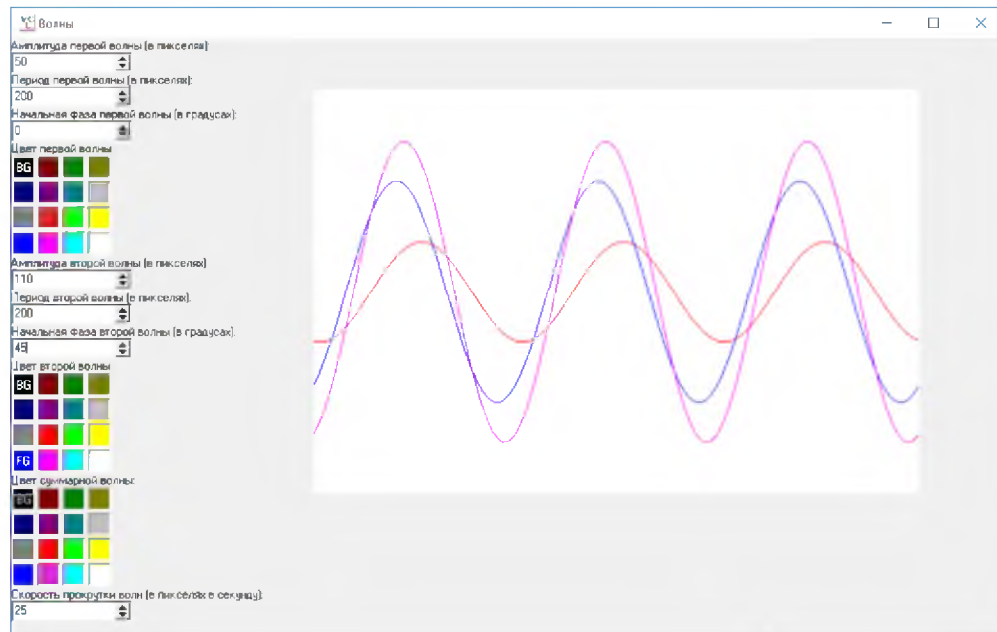


Рисунок 9 – Сложение колебаний с разностью фаз $\frac{\pi}{4}$

Сложение взаимно перпендикулярных гармонических колебаний в PascalABC с оживлением и возможностью изменения по алгоритму и получение фигур Лиссажу

Нам даны следующие величины: амплитуды первой и второй волн (A_1 , A_2), периоды колебаний первой и второй волн (T_1 , T_2), начальные фазы первой и второй волн (ϕ_1 , ϕ_2).

1. В программе PascalABC[7] загружаем нужные нам для работы модули и декларируем необходимые переменные:

```

program Lissazhu;
uses VCL, Utils;
var
MainForm: Form;
y, maxw: integer;
SE1, SE2, SE3, SE4, SE5, SE6, SE7: SpinEdit;
CG1, CG2, CG3: ColorGrid;
BM: Bitmap;
CT: CanvasType;
PB: PaintBox;
T: Timer;
first: boolean;

```

2. Записываем функции, которые упростят процесс добавления полей для ввода исходных числовых значений и для выбора цветов, а также процедуру, которая при каждом изменении значений форм нашего окна будет заново отрисовывать фигуру Лиссажу:

```

procedure OnChange;
begin
BM.Clear;
PB.Draw(0, 0, BM);
first:=true;
end;
function addText(s: string): TLabel;
begin
Result:=TLabel.Create(0, y, s);
if Result.Width>maxw then maxw:=Result.Width;
Inc(y, Result.Height);
end;
function addSpinEdit(i: integer): SpinEdit;
begin
Result:=SpinEdit.Create(0, y, IntToStr(i));
Result.OnChange:=OnChange;
if Result.Width>maxw then maxw:=Result.Width;
Inc(y, Result.Height);
end;

```



```
function addColorGrid(i: integer): ColorGrid;
begin
  Result:=ColorGrid.Create(0, y);
  Result.BackgroundEnabled:=false;
  Result.ForegroundIndex:=i;
  Result.OnClick:=OnChange;
  if Result.Width>maxw then maxw:=Result.Width;
  Inc(y, Result.Height)
end;
```

3. Прописываем функцию, вычисляющую координату x или y с учетом амплитуды, периода, начальной фазы волны и времени, прошедшего с начала запуска программы:

```
function f(A, T, fi: integer; s: real): integer;
begin
  Result:=trunc(A*sin(2*pi*s/T+fi*pi/180))
end;
```

В математическом виде эта функция выглядела бы так:

$$f = \left[A \cdot \sin \left(\frac{2\pi s}{T} + \frac{\varphi\pi}{180} \right) \right] \quad (1.5)$$

где A – амплитуда волны в пикселях, T – период колебаний волны в секундах, φ – начальная фаза волны, s – количество секунд, прошедших с момента запуска программы, квадратные скобки – целая часть от числа.

4. Прописываем процедуру прорисовки графика с использованием двойной буферизации (чтобы не было т.н. "моргания")[5]:

```
procedure Draw;
var A1, T1, fi1, A2, T2, fi2, v, x, y: integer; color1, color2, color3: ColorType; s: real;
begin
  A1:=SE1.Value; //считываем амплитуду первой волны из поля ввода
  T1:=SE2.Value; //считываем период колебаний первой волны из поля ввода
  fi1:=SE3.Value; //считываем начальную фазу первой волны из поля ввода
  //те же данные получаем о второй волне
  A2:=SE4.Value;
  T2:=SE5.Value;
  fi2:=SE6.Value;
  //получаем цвет суммарной волны
  color3:=CG3.ForegroundColor;
  with CT do
  begin
  with BMdo
  begin
  s:=Milliseconds/1000; //считываем кол-во секунд с момента компиляции программы
  //рисуем волну
  Pen.Color:=color3;
  x:=Width div 2+f(A1, T1, fi1, s);
  y:=Height div 2-f(A2, T2, fi2, s);
  if first then
  begin
  MoveTo(x, y);
  first:=false
  end
  else LineTo(x, y);
  //рисуем буфер в графическом окне
  PB.Draw(0, 0, BM)
  end
  end
  end;
```

5. Создаём основную программу:

```
begin
  //Создаём главную форму
  MainForm:=Form.Create('Волны');
  MainForm.Width:=1000;
  MainForm.Height:=600;
  y:=0;
  maxw:=0;
```

```
//добавляем в форму все необходимые поля
addText('Амплитуда первой волны (в пикселях):');
SE1:=addSpinEdit(200);
addText('Период первой волны (в секундах):');
SE2:=addSpinEdit(2);
addText('Начальная фаза первой волны (в градусах):');
SE3:=addSpinEdit(0);
addText('Амплитуда второй волны (в пикселях):');
SE4:=addSpinEdit(200);
addText('Период второй волны (в секундах):');
SE5:=addSpinEdit(1);
addText('Начальная фаза второй волны (в градусах):');
SE6:=addSpinEdit(0);
addText('Цвет суммарной волны:');
CG3:=addColorGrid(10);
//создаём буфер
BM:=Bitmap.Create(600, 400);
CT:=BM.Canvas;
//создаём графическое окно в форме
PB:=PaintBox.Create(maxw+50, 50, 600, 400);
first:=true;
//запускаем таймер, который будет каждые 100 миллисекунд вызывать процедуру вырисовки графика в графическом окне
T:=Timer.Create(Draw, 1)
end.
```

6. Запускаем программу, меняем некоторые параметры и через некоторое время видим такое изображение:

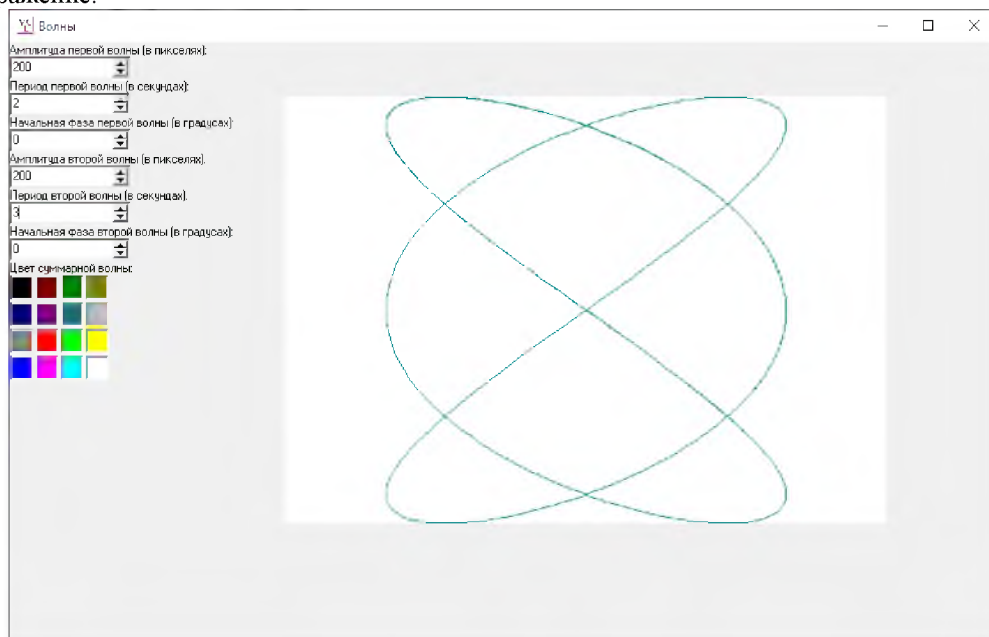


Рисунок 10 – Пример фигуры Лиссажу

Именно такая фигура Лиссажу отрисовывается при сложении двух взаимно перпендикулярных гармонических колебаний с отношением периодов 2:3 и разностью начальных фаз 0.

Сложение гармонических колебаний в Excel, без оживления и с возможностью изменения по алгоритму

Нам даны следующие величины: шаг времени волны (dt) [с], высоты первой и второй волн (A_1 , A_2) [м], частоты первой и второй волн (ω_1 , ω_2) [рад/с], углы первой и второй волн (φ_1 , φ_2) [градусы].

1. В табличном процессоре Microsoft Excel добавляем в первую ячейку на каждый столбец следующие величины: dt, A_1 , A_2 , ω_1 , ω_2 , φ_1 , φ_2 .
2. Под каждый заполненный столбец вводим следующие значения: 0,3; 1,5; 2,5; 3,14; 3,14; 30; 30 (см. рис. 11).

Буфер обмена		Шрифт		Выравни			
A1		fx		dt			
	A	B	C	D	E	F	G
1	dt	A ₁	A ₂	ω ₁	ω ₂	φ ₁	φ ₂
2	0,2	1,5	2,5	180	180	30	30

Рисунок 11 – Данные величины и их значения

3. Через Вставка/Таблица вставляем таблицу втретью ячейку первого столбца, расширяем её до столбца «D» и до ячейки «43».

4. Первый столбец таблицы называем t, второй – «1-ая волна», третий – «2-ая волна», четвертый – «Сумма волн».

5. Под ячейкой «t» вводим 0, далее в пятую ячейку столбца вводим функцию: =A4+\$A\$2. Включаем данную формулу для всех последующий ячеек в этом столбце.

6. Под ячейкой «1-ая волна» действует формула: $x = A_1 * \cos(\omega_1 * t + \varphi_1)$, по которой вводим функцию: =B\$2*COS(\$D\$2*A4+РАДИАНЫ(\$F\$2)), включаем автозаполнение всего столбца.

7. Совершаем аналогичную операцию со «2-ой волной» по формуле: $x = A_2 * \cos(\omega_2 * t + \varphi_2)$ с помощью функции: =C\$2*COS(\$E\$2*A4+РАДИАНЫ(\$G\$2)).

8. Под ячейкой «Сумма волн» по формуле сложения амплитуд: $A = A_1 + A_2$ столбец должен начинаться с функции: =СУММ(B4;C4). Амплитуда результирующего колебания зависит от разности фаз ($\varphi_2 - \varphi_1$) складываемых колебаний:

- Если $(\varphi_2 - \varphi_1) = \pm 2 * m * \pi$ ($m = 0, 1, 2, \dots$), то амплитуда результирующего колебания равна сумме амплитуд складываемых колебаний: $A = A_1 + A_2$;

- Если $(\varphi_1 - \varphi_2) = \pm (2 * m + 1) * \pi$ ($m = 0, 1, 2, \dots$), то амплитуда результирующего колебания равна разности амплитуд складываемых колебаний: $A = |A_1 - A_2|$ (см. рис. 12).

	A	B	C	D
3	t	1-ая волна	2-ая волна	Сумма волн
4	5	-1,30496931	-2,17494884	-3,4799181
5	5,2	-0,62143261	-1,03572102	-1,6571536
6	5,4	0,299237571	0,498729285	0,79796686
7	5,6	1,105721198	1,842868664	2,94858986
8	5,8	1,490270862	2,48378477	3,97405563
9	6	1,306145627	2,176909378	3,48305501
10	6,2	0,623606142	1,039343571	1,66294971
11	6,4	-0,29689623	-0,49482705	-0,7917233
12	6,6	-1,10410549	-1,84017581	-2,9442813
13	6,8	-1,48999732	-2,48332887	-3,9733262
14	7	-1,30731864	-2,17886439	-3,486183
15	7,2	-0,62577809	-1,04296349	-1,6687416
16	7,4	0,294554139	0,490923565	0,7854777
17	7,6	1,102486974	1,837478289	2,93996526
18	7,8	1,489719998	2,482866663	3,97258666
19	8	1,308488328	2,18081388	3,48930221
20	8,2	0,627948454	1,046580756	1,67452921
21	8,4	-0,2922113	-0,48701883	-0,7792301
22	8,6	-1,10086566	-1,83477611	-2,9356418
23	8,8	-1,4894389	-2,48239816	-3,9718371
24	9	-1,3096547	-2,18275784	-3,4924125
25	9,2	-0,63011722	-1,05019537	-1,6803126
26	9,4	0,289867719	0,483112865	0,77298058
27	9,6	1,099241563	1,832069272	2,93131083
28	9,8	1,489154018	2,481923364	3,97107738
29	10	1,310817753	2,184696254	3,49551401
30	10,2	0,632284393	1,053807322	1,68609172

Рисунок 12 – Таблица вычислений графиков

9. Для вывода получившихся графиков функций получаем диаграмму с двумя волнами и их суммой через «Вставка/Диаграммы/Объемный график» (см.рис. 13).

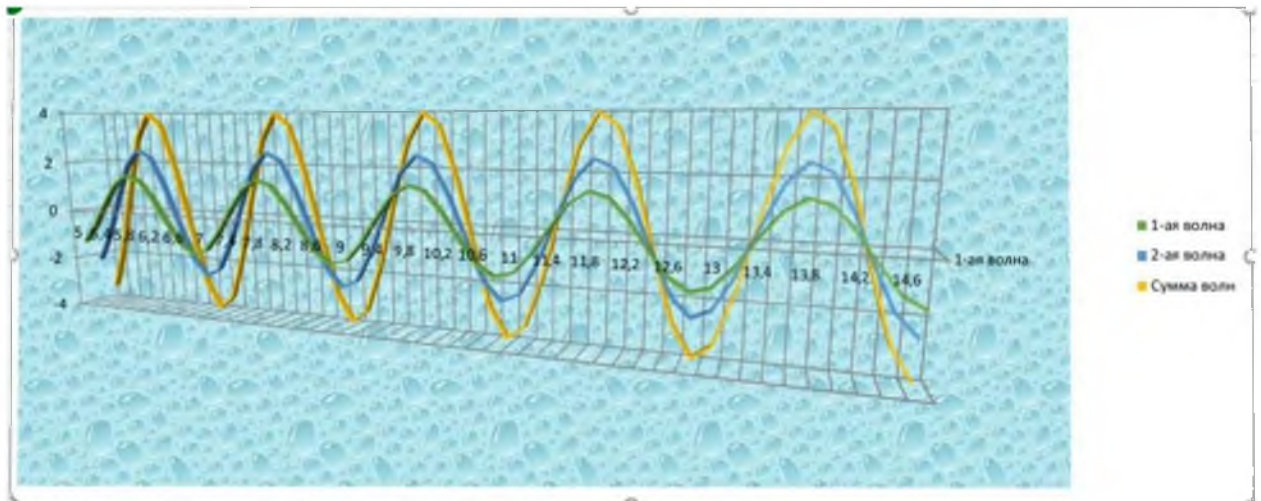


Рисунок 13 – Диаграмма полученных графиков

Оживление колебательных волн с помощью VBA

Наша волна будет двигаться вперед с помощью одной запрограммированной кнопки. Для этого нам понадобится:

1. Вставить кнопку в любое рабочее место в Excel «Разработчик/Вставка/Кнопка (элемент ActiveX).
2. В режиме конструктора два раза щелкнуть левой кнопкой мыши по кнопке. Откроется окно Microsoft Visual Basic, где мы и будем кодить.

```
3. Наш код должен выглядеть примерно так:
Private Sub Command_Bottom1_Click()
For R = 1 To 50
t = 0.1 * R
Cells(4, 1) = t
DoEvents
NextR
EndSub
```

4. Открываем окно Excel, выходим из режима конструктора и нажимаем на кнопку. Диаграмма оживает [1,6].

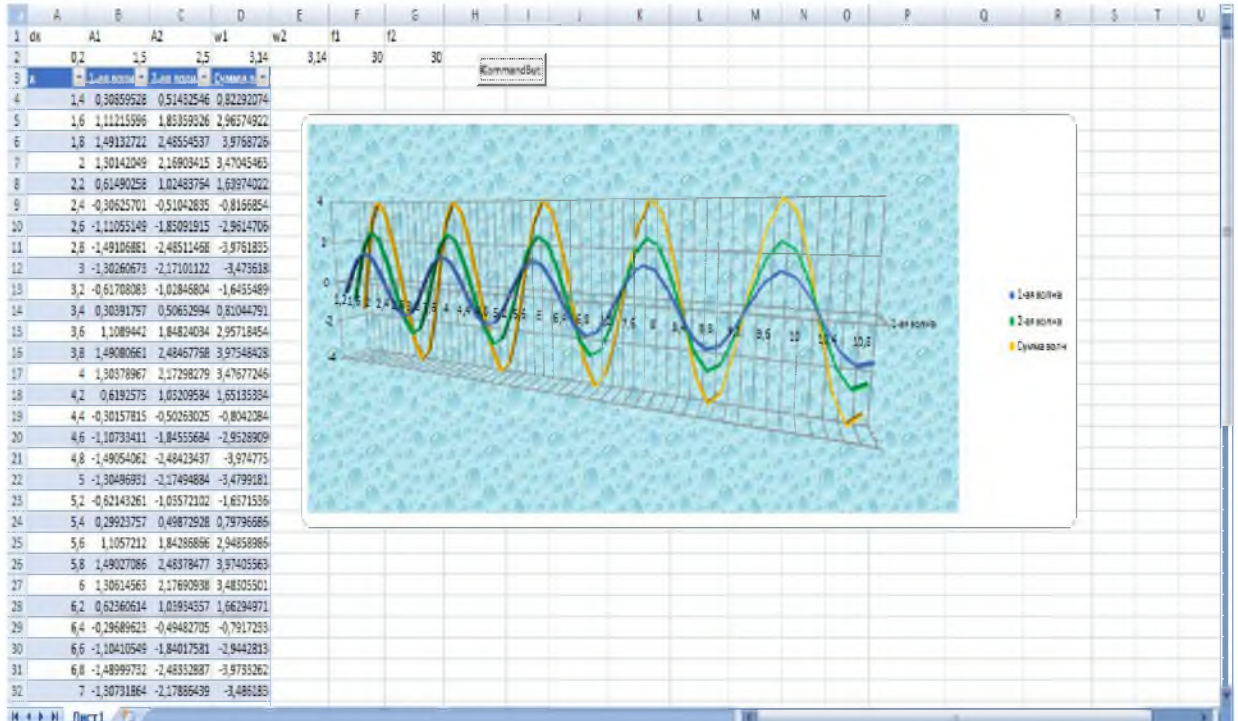


Рисунок 14 – Анимированный график (щелкните по нему два раза)

Данный алгоритм можно использовать для моделирования наложения волновых процессов как в любой упругой среде при изучении устойчивости судна на воде, конструкций, вибродиа-

гностики механизмов, так и для анализа интерференционных явлений в оптике и при распространении электромагнитных волн в радио- и оптической связи. Кроме того, методику можно

применять с целью визуализации моделирования сложения колебаний в учебном процессе.

Литература

1. Трофимова Т.И. Колебания и волны. Оптика. Квантовая физика. /Т.И. Трофимова, А.В. Фирсов. – М.: Дрофа, 2006. – 288 с.
2. Дубнищев Ю.Н. Колебания и волны /Ю.Н. Дубнищев. – СПб.: Лань, 2011. – 384 с.
3. Дукарт А.В. Задачи теории ударных гасителей колебаний/А.В. Дукарт. – М.: Издательство Ассоциации строительных вузов, 2006. – 208 с.
4. Гарбер Г.З. Основы программирования на VisualBasic и VBA в Excel 2007/Г.З. Гарбер. – М.: СОЛОН-ПРЕСС, 2008. – 192 с.
5. Двойная буферизация графики | MicrosoftDocs [Электронный ресурс]: Официальный сайт Microsoft. – URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/advanced/double-buffered-graphics/> (дата обращения 5.12.19)
6. Роман С. Использование Макросов в Excel/С. Роман. –СПб.: Издательский дом Питер, 2004. – 507 с.
7. Справка PascalABC.NET [Электронный ресурс]: Официальный сайт PascalABC.NET. – URL:<http://pascalabc.net/downloads/pabcnethelp/index.htm> (дата обращения 5.12.19)

References

1. Trofimova T.I. Kolebaniya i volny. Optika. Kvantovaya fizika./T.I. Trofimova, A.V. Firsov. – M.: Drofa, 2006. – 288 s.
2. Dubnishchev YU.N. Kolebaniya i volny/YU.N. Dubnishchev. – SPb.: Lan', 2011. – 384 s.
3. Dukart A.V. Zadachi teorii udarnyh gasitelej kolebanij/A.V. Dukart. – M.: Izdatel'stvo Associacii stroitel'nyh vuzov, 2006. – 208 s.
4. Garber G.Z. Osnovy programmirovaniya na VisualBasic i VBA v Excel 2007/G.Z. Garber. – M.: SOLON-PRESS, 2008. – 192 s.
5. Dvojnaya buferizaciya grafiki | MicrosoftDocs [Elektronnyj resurs]: Oficial'nyj sajt Microsoft. – URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/advanced/double-buffered-graphics/> (data obrashcheniya 5.12.19)
6. Roman S. Ispol'zovanie Makrosov v Excel/C. Roman. –Spb.: Izdatel'skij dom Piter, 2004. – 507 s.
7. Spravka PascalABC.NET [Elektronnyj resurs]: Oficial'nyj sajt PascalABC.NET. – URL:<http://pascalabc.net/downloads/pabcnethelp/index.htm> (data obrashcheniya 5.12.19)