2. Марухина О. В. Технологии визуализации результатов экспериментальных исследований / О. В. Марухина, О. Г. Берестнева, В. А. Воловоденко, К. А. Шаропин // Информационные и математические технологии в науке и управлении. — Ч. 3. — Иркутск, 2010. — С. 165-171.

3. Попечителев Е. П. Анализ числовых таблиц в биотехнических системах обработки экспериментальных данных / Е. П. Попечителев. — Л. : Наука, 1985. — 148 с.

4. Шаропин К. А. Визуализация результатов экспериментальных исследований / К. А. Шаропин, О. Г. Берестнева, Г. И. Шкатова // Известия Томского политехнического университета. — 2010. — Т. 316. — № 5. — С. 172-176.

5. Wong D. M. The Wall Street Journal Guide to Information Graphics: The Dos and Don'ts of Presenting Data, Facts, and Figures / Dona M. Wong. — New York : W.W. Norton & Company, 2010. — 160 p.

6. Берестнева О. Г. Визуализация экспериментальных многомерных данных на основе обобщенных графических образов / О. Г. Берестнева, В. А. Воловоденко, К. А. Шаропин, О. М. Гергет // Вестник науки Сибири. — 2011. — № 1(1). — С. 363-369.

7. Толок А. В. Модель геометрической поддержки для описания формы поверхности / А. В. Толок, В. В. Мухин // Сборник научных трудов, посвященных 10-летию университета. Математика. Физика. — Запорожье : Запорожский государственный университет, 1995. — С. 86-91.

8. Толок А. В. Алгоритм итерационного уточнения области исследования поверхности / А. В. Толок, В. В. Мухин // Вісник Запорізького державного університету. — 1998. — № 2. — С. 90-97.

9. Толок А. В. Исследование функции одной переменной с помощью графических образов / А. В. Толок, В. В. Мухин // Вісник Запорізького державного університету. — 1999. — № 1. — С. 108-112.

10. Толок А. В. Рекурсивный алгоритм разбиения области с дополнительными параметрами уточнения / А. В. Толок, В. В. Мухин // Вестник Херсонского государственного университета. — 2003. — № 3(19). — С. 312-314.

11. Самарский А. А. Введение в численные методы / А. А. Самарский. — М. : Наука, 1987. — 288 с.

12. Рвачев В. Л. Теория R-функций и некоторые ее приложения / В. Л. Рвачев. — К. : Наукова думка, 1982. — 552 с.

13. Рвачев В. Л. Введение в теорию R-функций / В. Л. Рвачев, Т. И. Шейко // Проблемы машиностроения. — 2001. — Т. 4, № 1-2. — С. 46-58.

УДК 004.9

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ВИЗУАЛИЗАЦИИ ПОВЕРХНОСТИ ФУНКЦИЙ ДВУХ ПЕРЕМЕННЫХ

Науменко Д. А., бакалавр, Кондратьева Н. А., к. ф.-м. н., доцент, Мухин В. В., к. т. н., доцент, Леонтьева В. В., к. ф.-м. н., доцент

Запорожский национальный университет

В статье приведён способ организации графической информации, который даёт возможность выделить особые точки и линии изгибов поверхности. Разработана реализация алгоритма уточнения рассматриваемой области для исследования поверхности.

Ключевые слова: функция двух переменных, графическая информация, программа на С++, особые точки поверхности, линии изгибов поверхности.

Науменко Д. А., Кондрат'єва Н. А., Мухін В. В., Леонтьєва В. В. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВІЗУАЛІЗАЦІЇ ПОВЕРХНІ ФУНКЦІЙ ДВОХ ЗМІННИХ / Запорізький національний університет, Україна

У статті наведено спосіб організації графічної інформації, який дає можливість виділити особливі точки та лінії вигинів поверхні. Розроблена реалізація алгоритму уточнення розглянутої області для дослідження поверхні.

Ключові слова: функція двох змінних, графічна інформація, програма на С++, особливі точки поверхні, лінії вигинів поверхні.

Naumenko D. A., Kondratyeva N. A., Mukhin V. V., Leontieva V. V. INFORMATION TECHNOLOGY RENDER THE SURFACE OF A FUNCTIONS OF TWO VARIABLES / Zaporizhzhya National University, Ukraine

In this paper, proposed a new way of organizing information graphics, which allows to allocate specific points and lines bends surface of the object, as well as to provide a reliable and fair graphical representation of geometric objects, that is especially important in the design of complex engineering structures. The proposed method is based on the recursive refinement of the area under consideration for studying the surface of a geometric object. It is considered recursive algorithm for visualization differential properties of functions of two variables. The proposed algorithm allows to visualize the properties of surfaces such as the form and wave front with maximum precision and is associated with a dynamic process. It gives the possibility of operational interference in the processing in each level in the hierarchy of recursion, in turn, permits variation of the methods of investigation in local areas of considered area. The advantage of the proposed algorithm is the possibility of visual evaluation of the image still in the early stage of processing the graphic information. Implemented in the algorithm binding gradation of tone to the direction of projection of the normal vector to the plane *XOY* can greatly simplify the perception of the direction of the surface slope, while intensifying the points in which this line is missing. Such points in this case are the extreme points at which the tangent plane is parallel to the horizon. This approach allows to allocate character of breaks investigated surface and therefore defined as «***form***». In this approach, the concept of Form determined by the angle between the projection of the gradient on the plane *OXY* and the positive direction of the axis *OX*.

The second property, under consideration, characterizes the deviation of the normal to the selected direction and activates perception changing nature of the surface, ignoring the nature of the fracture. In this case, decided to determine it as the «***wavefront***». At the same time, introduced the concept of Wavefront is defined as the angle between the gradient and the axis *OZ*.

The proposed information visualization technology, allows to extend the class of problems for the visual analysis in different applications of mathematics.

Key words: the function of two variables, graphics, program in C++, singular points of surfaces, lines of curved surfaces.

ВВЕДЕНИЕ

В наше время очень интенсивно развивается программное обеспечение [1-3]. Одним из самых популярных направлений является разработка новых технологий и методов визуализации результатов работы программ, то есть такой их подачи, в которой они наиболее адекватно воспринимаются человеком [1, 2].

Понимание графического образа как графической информации является актуальной проблемой. Для ее решения необходимо разработать новые принципы. Они должны позволять активизировать информацию, которая была интересной, за счет использования различных изобразительных средств [1, 3, 5].

Визуальный анализ работает с ***графической информацией***, которая представляет собой визуальный образ геометрического объекта и сводится к зрительному выделению основных свойств поверхности [1, 2]. Традиционные изображения поверхности не всегда позволяют однозначно определить особые точки и линии изгибов формы поверхности. Необходима дополнительная оценка с применением зрительной интерполяции для определения элементов подобного рода [1].

Решению этой проблемы посвящена предлагаемая работа, в которой изложен новый способ организации графической информации. Он дает возможность отображать характеристики графической информации и основан на рекурсивном уточнении рассматриваемой области для исследования поверхности геометрического объекта. Программная реализация выполнена на языке программирования C++ с использованием библиотек WinForms.

РЕАЛИЗАЦИЯ РЕКУРСИВНОГО РАЗБИЕНИЯ

В алгоритме рекурсивного разбиения прямоугольная область, заданная пользователем, на которой исследуется функция, разбивается на участки в виде правильных n-угольников. Правильность участка обусловлена использованием радиального принципа разбиения области на подобласти.

Одной из проблем, возникающих при работе подобного алгоритма, является определение места расположения n-угольника относительно заданной области исследования функции [1-4]. В данной работе эта проблема решается рекурсивным образом, сущность которого состоит в следующем. На первом шаге рекурсии определяется радиус R и координаты X_c, Y_c центра окружности, охватывающей прямоугольную область исследования функции с вписанным в нее правильным n-угольником. На каждом следующем шаге рекурсии участок в виде правильного n-угольника разбивается на n одинаковых секторов. При этом рассматриваются наклонные плоскости, которые образуются участками подобластей внутри области как приближенные касательные плоскости в окрестности точки P. Сама точка P при подобном рассмотрении приближенной касательной плоскости не имеет, что потребовало усовершенствовать алгоритм, добавив в работу алгоритма дополнительную подобласть той же

конфигурации, что и остальные, но расположенную в центре подобласти. Таким образом, рекурсия продолжается до тех пор, пока это либо необходимо исследователю, либо до пиксельного уровня.

Время, затрачиваемое на получение графического образа с помощью рекурсивного алгоритма, зависит от двух параметров: количества секторов поворота n и степени рекурсивной вложенности k. Процедура сортировки на очередном шаге рекурсии выполняется не более, чем $(n^k - 1)/(n-1) - 1$ раз, где n – количество секторов, а k – глубина рекурсии.

Следует отметить, что данный подход позволяет получить измененное представление исследуемого графического образа уже на первых шагах рекурсии. Этот факт дает возможность сократить время реакции исследователя на поиск различного рода ошибок, возникающих при описании исследуемого объекта.

На языке программирования C++ данный алгоритм выглядит следующим образом (см. рис.1):

```
void RectangleDraw(double Current,double Count,double R,double Xc,double Yc,double m){
    Form1->StatusBar1->Panels->Items[5]->Text=R;
    if  (Current==Count)
{
      if((Xc - R>= X1) &&( Xc + R <=X2 )&&  (Yc - R>=Y1) && (Yc + R<=Y2) )
  {
   // Определение значений функции в заданных узловых точках рассматриваемой подобласти
   // для нужной задачи
      TPoint A[100];
      double angle=0;
      int ii;
      for (ii=0;ii<=m;ii++)
      {
            A[ii].x=int(Xc+r1*cos((angle*M_PI/180+M_PI/m)));
            A[ii].y=int(Yc+r1*sin((angle*M_PI/180+M_PI/m)));
            angle=angle+360/m;
      }
      for (ii=0;ii<=m;ii++)   Form1->Image1->Canvas->Polygon(A,m);
      return;
                  }
  }
 else { return; }
}
else
{
      for (int i = 0;i < m; i++)
      RectangleDraw (Current+1,Count, R/2.,Xc + R/2.*cos(i*2*M_PI/m), Yc +R/2.*sin(i*2*M_PI/m),m);
      RectangleDraw (Current+1,Count, R/2.,Xc , Yc , m);
}
   return;
}
```

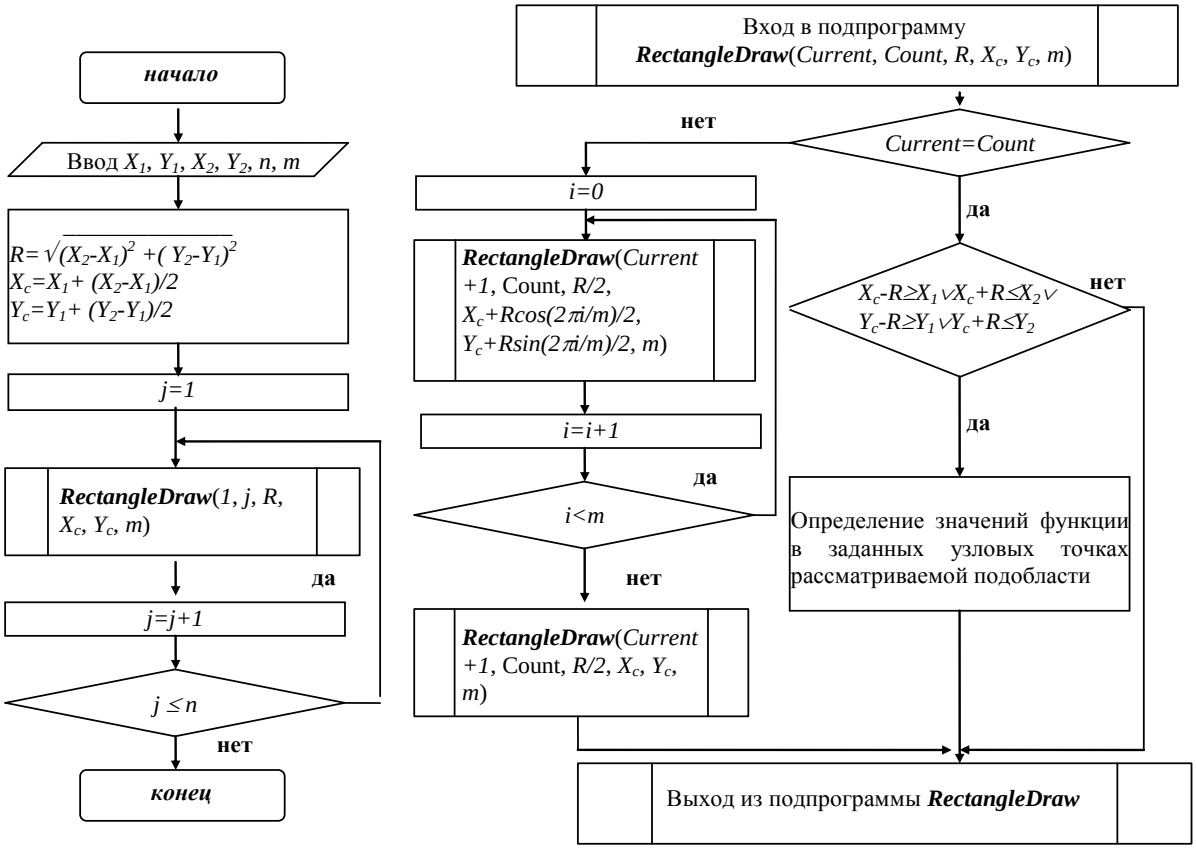


Рис. 1. Блок-схема рекурсивного алгоритма

ПОСТРОЕНИЕ ВИЗУАЛЬНОГО ОБРАЗА ТРЕХ ХАРАКТЕРИСТИК ПОВЕРХНОСТИ

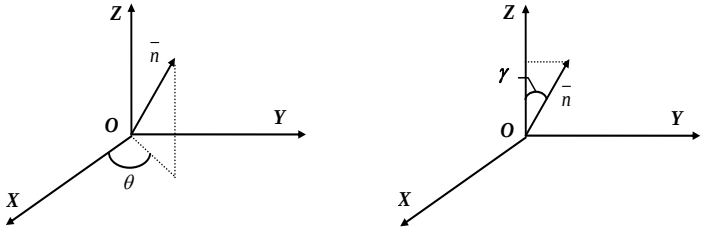
В данной статье в качестве примера реализации рекурсивного разбиения приведено исследование графика функции двух переменных и определение его свойств, таких как ***форма***, ***фронт волны*** и ***линии уровня.*** Остановимся на данных свойствах более подробно.

Привязка градации тона к направлению проекции вектора нормали на плоскость XOY позволяет упростить восприятие направления склонов поверхности, активизируя при этом точки, в которых это направление отсутствует. Такими точками являются экстремальные точки, в которых касательная плоскость параллельна горизонту. Такой подход позволяет выделить характер изломов поверхности, поэтому определен как «***форма***». При этом форма определяется углом между проекцией градиента на плоскость OXY и положительным направлением оси OX .

Второе свойство характеризует определение отклонения нормали от выбранного направления и активизирует восприятие изменения характера поверхности, пренебрегая характером излома. В этом случае решено определить его как «***фронт волны***». При этом фронт волны определяется как угол между градиентом и осью OZ .

Если под ***градиентом*** в точке поверхности понимать вектор нормали касательной плоскости в этой точке, то необходимо определить:

- средний угол между направлением проекции градиента на плоскость OXY и осью OX (рис. 2 а));
- средний угол между градиентом и осью OZ (рис. 2 б)).



а) б)

Рис. 2: а) θ – характеризует форму поверхности; б) γ – характеризует фронт волны

При этом под ***средним углом*** будем понимать среднеарифметическое значение соответствующих углов θ и γ для ΔABC и $\Delta A_1B_1C_1$ (рис. 2), которые обозначим θ_{cp} и γ_{cp}.

Таким образом, задача сводится к определению двух углов θ_{cp} и γ_{cp}.

Для каждого треугольника записываются коэффициенты a, b, c, d уравнения $ax+by+cz=d$ плоскости, в которой расположен треугольник. В общем случае коэффициенты a, b, c, d вычисляются по координатам вершин треугольника.

Алгоритм определения «***формы***» следующий:

1. Вычисляем координаты вершин ΔABC (см. рис. 3):

$$x_s = x_c + R\cdot\cos\left(\frac{(2s-p)\pi}{3}\right),$$
$$y_s = y_c + R\cdot\sin\left(\frac{(2s-p)\pi}{3}\right), \tag{1}$$

где $p=0$ для ΔABC, $s=0,1,2$, R – радиус окружности, описанной вокруг рассматриваемого участка в виде правильного n-угольника, x_c, y_c – координаты центра, определяющие параметр положения исследуемого участка.

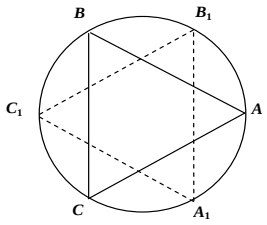


Рис. 3. Схема расположения треугольников на рассматриваемом участке поверхности

2. Вычисляем значения z исследуемой функции в виде

$$z_s = f(x_s, y_s),\ s=0,1,2, \tag{2}$$

где x_s, y_s – координаты вершин треугольника.

3. Если

$$(x_1-x_0)\cdot(y_2-y_0)-(x_2-x_0)\cdot(y_1-y_0)=V, \tag{3}$$

где (x_0,y_0), (x_1,y_1), (x_2,y_2) – координаты вершин полученного треугольника, не равно нулю, т.е. три точки не лежат на одной прямой, то переходим к следующему шагу алгоритма; иначе, плоскость ΔABC параллельна оси OZ и проекция нормали с осью OX составляет угол $\theta=0$ и в этом случае необходимо осуществить переход к шагу 7.

4. Вычисляем коэффициенты a, b, c плоскости ΔABC в виде:

$$\begin{aligned} a &= yA(zB-zC)-yB(zA-zC)+yC(zA-zB);\\ b &= -(xA(zB-zC)-xB(zA-zC)+xC(zA-zB));\\ c &= xA(yB-yC)-xB(yA-yC)+xC(yA-yB). \end{aligned} \tag{4}$$

5. Вычисляем угол θ между проекцией нормали и осью OX.

Если одно из условий

а) $\frac{a}{V}=0$ и $\frac{b}{V}=0$, то переходим к шагу 8;

б) $\frac{a}{V}=0$, то $\theta=\frac{\pi}{2}$;

в) $\frac{a}{V}>0$ и $\frac{b}{V}=0$, то $\theta=0$;

г) $\frac{a}{V} < 0$ и $\frac{b}{V} = 0$, то $\theta = \pi$;

выполняется, то переходим к шагу 7.

Иначе, вычисляем угол θ согласно формуле

$$\theta = arctg\frac{b}{a}. \qquad (5)$$

6. Возможны следующие случаи условий:

– $\frac{a}{V} > 0$ и $\frac{b}{V} < 0$, то $\theta = -\theta$;

– $\frac{a}{V} < 0$ и $\frac{b}{V} > 0$, то $\theta = \pi + \theta$;

– $\frac{a}{V} < 0$ и $\frac{b}{V} < 0$, то $\theta = \pi - \theta$.

7. Угол между проекцией нормали к плоскости ΔABC на плоскость OXY и осью OX равен $\theta_p = \theta$.

8. Вычисляем координаты вершин $\Delta A_1B_1C_1$ (см. рис. 3) для $p = 1$.

Последовательно выполняем с 1 – 7 шаги алгоритма для плоскости $\Delta A_1B_1C_1$.

9. Определяем средние углы для нормали к срединной плоскости в виде

$$\theta_{cp} = \frac{\theta + \theta_1}{2}.$$

Представим программный код реализации изложенного выше алгоритма.

```
double forma(double Xc1,double Yc1,double R1){
    double x01,x02,x03,y01,y02,y03,x11,x12,x13,y11,y12,y13;
    double z01,z02,z03,z11,z12,z13;
    double v,a,b,c,o,o1,o2;
    Xc1=Xc1*Kx+Xmin;
    Yc1=Yc1*Ky+Ymin;
    R1=R1*Kx;
    x01=Xc1+R1*cos(((2*0-0)*M_PI)/3);
    y01=Yc1+R1*sin(((2*0-0)*M_PI)/3);
    x02=Xc1+R1*cos(((2*1-0)*M_PI)/3);
    y02=Yc1+R1*sin(((2*1-0)*M_PI)/3);
    x03=Xc1+R1*cos(((2*2-0)*M_PI)/3);
    y03=Yc1+R1*sin(((2*2-0)*M_PI)/3);
    z01=fun(x01/Kx,y01/Ky);
    z02=fun(x02/Kx,y02/Ky);
    z03=fun(x03/Kx,y03/Ky);
    v=(x02-x01)*(y03-y01)-(x03-x01)*(y02-y01);
    o1=0;
if (v!=0) {
  a=y01*(z02-z03)-y02*(z01-z03)+y03*(z01-z02);
  b=-(x01*(z02-z03)-x02*(z01-z03)+x03*(z01-z02));
```

```
  c=x01*(y02-y03)-x02*(y01-y03)+x03*(y01-y02);
  if ((a==0)&&(b==0))
  if (a==0)              {o1=M_PI/2;}
  if (((a/v)>0)&&(b==0))   {o1=0;}
  if (((a/v)<0)&&(b==0))   {o1=M_PI;}
  if (((a/v)>0)&&((b/v)<0))  {o1=-o1;}
  if (((a/v)<0)&&((b/v)>0))  {o1=o1+M_PI;}
  if (((a/v)<0)&&((b/v)<0))  {o1=M_PI-o1;}
  if (a!=0)              { o1=atan(b/a);  }
}
      x11=Xc1+R1*cos(((2*0-1)*M_PI)/3);
      y11=Yc1+R1*sin(((2*0-1)*M_PI)/3);
      x12=Xc1+R1*cos(((2*1-1)*M_PI)/3);
      y12=Yc1+R1*sin(((2*1-1)*M_PI)/3);
      x13=Xc1+R1*cos(((2*2-1)*M_PI)/3);
      y13=Yc1+R1*sin(((2*2-1)*M_PI)/3);
      z11=fun(x11,y11);
      z12=fun(x12,y12);
      z13=fun(x13,y13);
      v=(x12-x11)*(y13-y11)-(x13-x11)*(y12-y11);
      o2=0;
if (v!=0) {
  a=y11*(z12-z13)-y12*(z11-z13)+y13*(z11-z12);
  b=-(x11*(z12-z13)-x12*(z11-z13)+x13*(z11-z12));
  c=x11*(y12-y13)-x12*(y11-y13)+x13*(y11-y12);
  if ((a==0)&&(b==0))
  if (a==0)              {o2=M_PI/2;}
  if (((a/v)>0)&&(b==0))   {o2=0;}
  if (((a/v)<0)&&(b==0))   {o2=M_PI;}
  if (((a/v)>0)&&((b/v)<0))  {o2=-o2;}
  if (((a/v)<0)&&((b/v)>0))  {o2=o1+M_PI;}
  if (((a/v)<0)&&((b/v)<0))  {o2=M_PI-o2;}
  if (a!=0)              {o2=atan(b/a);  }
}
  o=(o1+o2)/2;
return o;
}
```

Алгоритм описан.

Алгоритм определения «***фронт волны***» следующий:

1. Вычисляем координаты вершин ΔABC по формуле (1).

2. Вычисляем с помощью (2) значения z исследуемой функции.

3. Если условие (3) не равно нулю, то переходим к следующему шагу алгоритма; иначе, плоскость ΔABC параллельна оси OZ и проекция нормали с осью OZ составляет угол, $\gamma = 0$ и переходим к шагу 7.

4. Вычисляем с помощью формул (4) коэффициенты a, b, c плоскости ΔABC.

5. Вычисляем угол θ.

Если одно из условий

– $\frac{a}{V} = 0$ и $\frac{b}{V} = 0$, то $\theta = 0$;

– $\frac{a}{V} = 0$, то $\theta = \frac{\pi}{2}$;

выполняется, то переходим к шагу 6. Иначе, вычисляем угол θ согласно формуле (5).

6. Угол между нормалью к плоскости ΔABC и осью OZ определяется в виде

$$\gamma_p = arctg\left(\frac{a}{V}\cdot\cos(\theta) + \frac{b}{V}\cdot\sin(\theta)\right).$$

7. Вычисляем координаты вершин $\Delta A_1B_1C_1$ (см. рис. 3) для $p = 1$.

Последовательно выполняем с 1 – 6 шаги алгоритма для плоскости $\Delta A_1B_1C_1$.

8. Определяем средние углы для нормали к срединной плоскости в виде

$$\gamma_{cp} = \frac{\gamma + \gamma_1}{2}.$$

Представим программный код реализации изложенного выше алгоритма.

```
double volna(double Xc1,double Yc1,double R1){
    double x01,x02,x03,y01,y02,y03,x11,x12,x13,y11,y12,y13;
    double z01,z02,z03,z11,z12,z13;
    double v,a,b,c,o,o1,o2,ygol,ygol1,ygol2;
    Xc1=Xc1*Kx+Xmin;
    Yc1=Yc1*Ky+Ymin;
    R1=R1*Kx;
    x01=Xc1+R1*cos(((2*0-0)*M_PI)/3);
    y01=Yc1+R1*sin(((2*0-0)*M_PI)/3);
    x02=Xc1+R1*cos(((2*1-0)*M_PI)/3);
    y02=Yc1+R1*sin(((2*1-0)*M_PI)/3);
    x03=Xc1+R1*cos(((2*2-0)*M_PI)/3);
    y03=Yc1+R1*sin(((2*2-0)*M_PI)/3);
    z01=fun(x01/Kx,y01/Ky);
    z02=fun(x02/Kx,y02/Ky);
    z03=fun(x03/Kx,y03/Ky);
    v=(x02-x01)*(y03-y01)-(x03-x01)*(y02-y01);
    ygol1=0;
if (v!=0) {
  a=y01*(z02-z03)-y02*(z01-z03)+y03*(z01-z02);
  b=-(x01*(z02-z03)-x02*(z01-z03)+x03*(z01-z02));
  c=x01*(y02-y03)-x02*(y01-y03)+x03*(y01-y02);
```

```
  if (a!=0)              {o1=atan(b/a);  }
  if ((a==0)&&(b==0))  {  o1=0; }
  if (a==0)  {  o1=M_PI/2;    }
  ygol1=atan((a/v)*cos(o1)+(b/v)+sin(o1));
}
      x11=Xc1+R1*cos(((2*0-1)*M_PI)/3);
      y11=Yc1+R1*sin(((2*0-1)*M_PI)/3);
      x12=Xc1+R1*cos(((2*1-1)*M_PI)/3);
      y12=Yc1+R1*sin(((2*1-1)*M_PI)/3);
      x13=Xc1+R1*cos(((2*2-1)*M_PI)/3);
      y13=Yc1+R1*sin(((2*2-1)*M_PI)/3);
      z11=fun(x11,y11);
      z12=fun(x12,y12);
      z13=fun(x13,y13);
      v=(x12-x11)*(y13-y11)-(x13-x11)*(y12-y11);
      ygol2=0;
if (v!=0) {
  a=y11*(z12-z13)-y12*(z11-z13)+y13*(z11-z12);
  b=-(x11*(z12-z13)-x12*(z11-z13)+x13*(z11-z12));
  c=x11*(y12-y13)-x12*(y11-y13)+x13*(y11-y12);

  if (a!=0)              {o2=atan(b/a);  }
   if ((a==0)&&(b==0))  {  o1=0; }
  if (a==0)  {  o1=M_PI/2;    }
  ygol2=atan((a/v)*cos(o2)+(b/v)+sin(o2));
}
  ygol=(ygol1+ygol2)/2;
return ygol;
}
```

Алгоритм описан.

РЕЗУЛЬТАТЫ

Предложенные алгоритмы позволяют построить дифференциальные характеристики поверхностей, заданных аналитически.

При этом для реализации предложенных в работе алгоритмов использовался компилятор Borland C++ Builder 6.0 с выставленной опцией оптимизации кода.

Результаты работы реализованных алгоритмов сведены в таблицы результатов по соответствующей исследуемой функции.

Для проведения сравнительного анализа полученных в ходе проводимого исследования результатов, рассмотрим следующие примеры применения построенных алгоритмов.

1. Построим графики свойств для функции вида

$$F(x,y)=\frac{1}{5}\sin x\cos y-\frac{3}{2}\cos\left(7\frac{(x-\pi)^2+(y-\pi)^2}{4}\right)e^{-\left((x-\pi)^2+(y-\pi)^2\right)}$$

на области применения $x\in(0,2\pi)$; $y\in(0,2\pi)$.

При помощи математического пакета Mathcad получим для сравнения трёхмерный график функции $F(x,y)$ (рис.4).

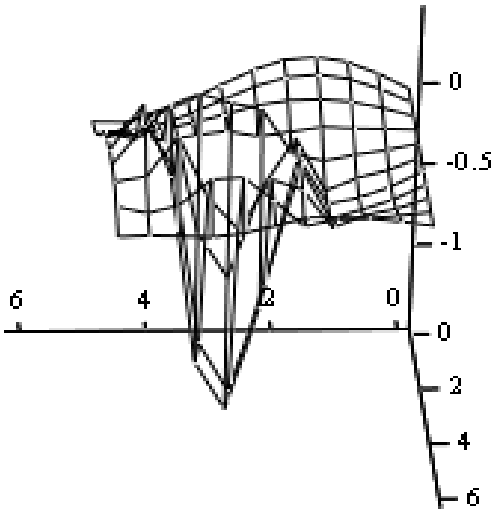
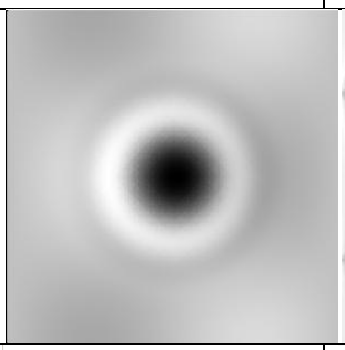
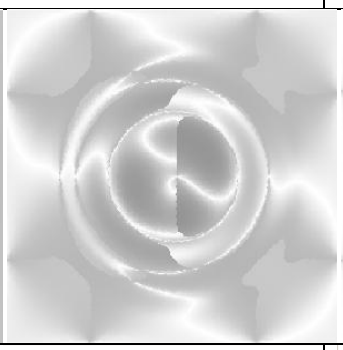
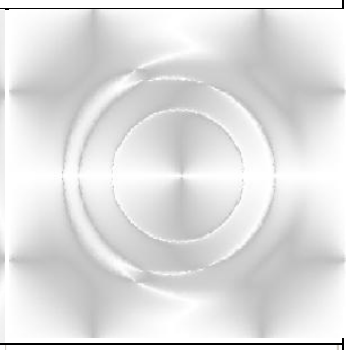
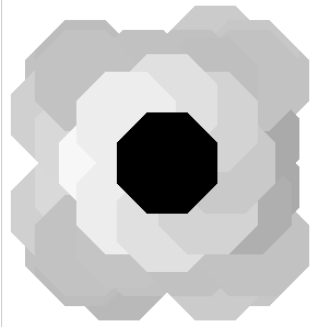
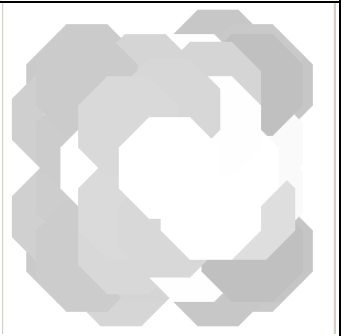
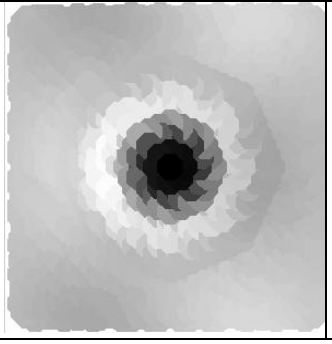
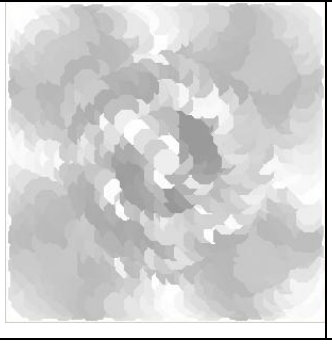
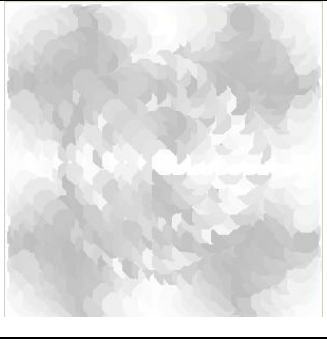


Рис. 4

Для исследуемой функции $F(x,y)=\frac{1}{5}\sin x\cos y-\frac{3}{2}\cos\left(7\frac{(x-\pi)^2+(y-\pi)^2}{4}\right)e^{-\left((x-\pi)^2+(y-\pi)^2\right)}$ приведем в табличном виде результаты выполнения описанных алгоритмов (табл.1.).

Таблица 1 – Результаты выполнения алгоритмов для функции $F(x,y)$

	Линии уровня	Форма	Фронт волны
Кол-во углов=4 Кол-во итераций=8			
Кол-во углов=8 Кол-во итераций=2			
Кол-во углов=8 Кол-во итераций=4			

2. Построим графики свойств для функции вида

$$G(x,y)=x+y+\cos\left(x^2+y^2\right)$$

на области применения $x\in(0,2\pi)$; $y\in(0,2\pi)$.

При помощи математического пакета Mathcad получим для сравнения трёхмерный график указанной функции $G(x,y)$ (рис. 5).

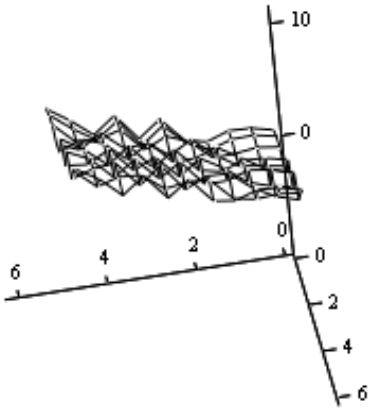
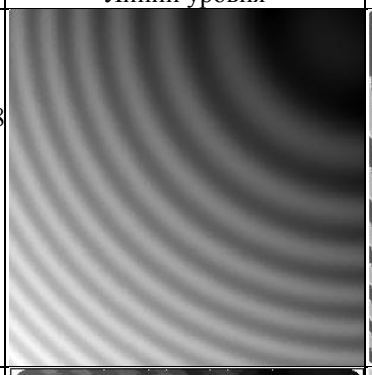
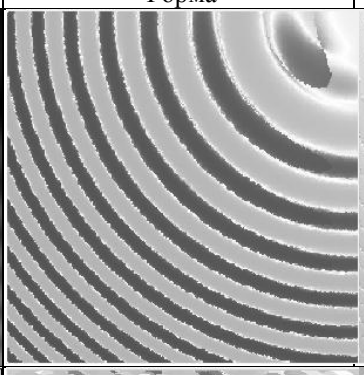
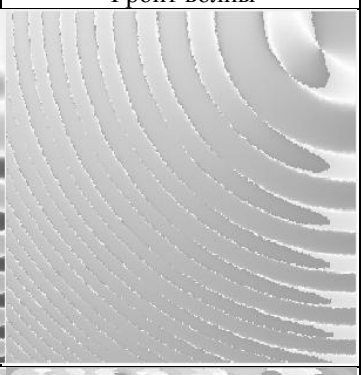
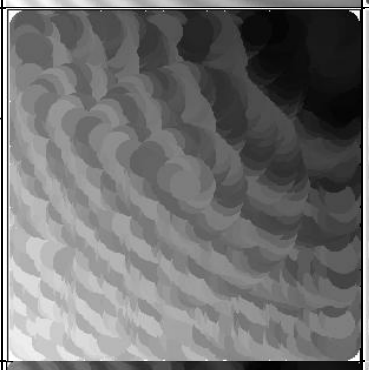
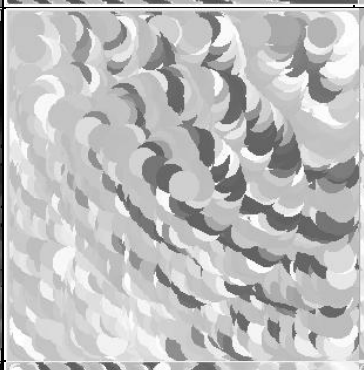
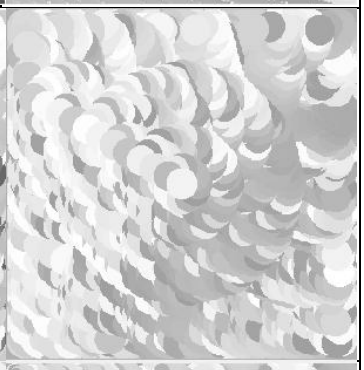
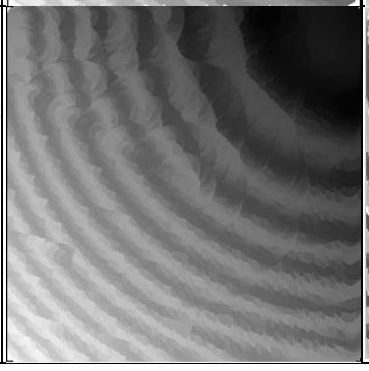
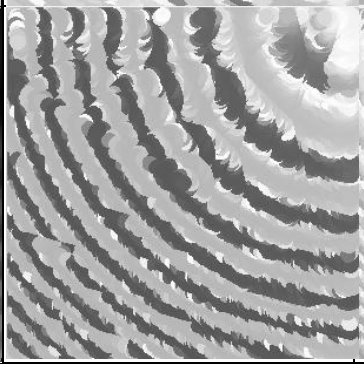
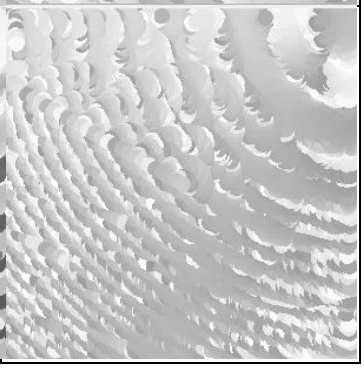


Рис. 5

Для исследуемой функции $G(x,y)=x+y+\cos\left(x^2+y^2\right)$ приведем в табличном виде результаты выполнения описанных алгоритмов (табл. 2.).

Таблица 2 – Результаты выполнения алгоритмов для функции $G(x,y)$

	Линии уровня	Форма	Фронт волны
Кол-во углов=4 Кол-во итераций=8			
Кол-во углов=12 Кол-во итераций=4			
Кол-во углов=12 Кол-во итераций=5			

Сравнивая полученные изображения для рассмотренных соответствующих функций вида $F(x,y)=\frac{1}{5}\sin x\cos y-\frac{3}{2}\cos\left(7\frac{(x-\pi)^2+(y-\pi)^2}{4}\right)e^{-\left((x-\pi)^2+(y-\pi)^2\right)}$ и $G(x,y)=x+y+\cos\left(x^2+y^2\right)$ (см. табл. 1, 2 соответственно), видно, что «общий» вид графика свойства поверхности можно заметить уже на первых итерациях.

Визуализация свойства *Форма* наглядно отображает все экстремальные точки независимо от значения функции в этих точках, поскольку изображение исключает такие характеристики поверхности как угол между градиентом и осью OZ. Это позволяет наблюдать за развитием волновых деформаций поверхности.

Свойство *Фронт волны* отражает относительную крутизну распространяемых волн, опуская при этом характеристику форма поверхности.

ЛИТЕРАТУРА

1. Толок О. В. Принципи візуальної підтримки властивостей поверхні на основі реконструкції реалістичного образу / О. В. Толок // Збірник наукових праць. Вісник Запорізького державного університету. Фізико-математичні науки, біологічні науки. — Запоріжжя, Запорізький національний університет. — 1998. — С. 76-79.
2. Толок А. В. Визуализация некоторых дифференциальных свойств поверхности на основе реконструкции реалистичного образа / А. В. Толок, В. В. Мухин // XXV Юбилейная международная конференция и дискуссионный клуб IT+SE’98 «Новые информационные технологии в науке, образовании, телекоммуникации и бизнесе». (Ч. 1. Украина, Крым, Ялта–Гурзуф, 15-24 мая). — 1998. — С. 162-164.
3. Вирт Н. Алгоритмы и структуры данных / Н. Вирт. — М. : Мир, 1989. — 360 с.
4. Канторович Л. В. Приближенные методы высшего анализа / Л. В. Канторович, В. И. Крылов. — М.-Л. : Физматгиз, 1962. — 708 с.
5. Макаров Е. Г. Инженерные расчеты в Mathcad 15 : Учебный курс / Е. Г. Макаров. — СПб. : Питер, 2011. — 400 с.

УДК 531.36

МОДЕЛЮВАННЯ МАЙЖЕ КОНСЕРВАТИВНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ ЗВОРОТНОГО ЗВ’ЯЗКУ

[1]Новицький В. В., д. ф.-м. н., професор, [1]Коломійчук О. П., к. ф.-м. н., [2]Святовець І. Ф.

[1]Ін-т математики НАН України

[2]Запорізька державна інженерна академія

Пропонується один з можливих підходів визначення умов побудови майже консервативної системи за допомогою зворотного зв’язку. Наводяться приклади знаходження керування для побудови майже консервативної системи.

Ключові слова: майже консервативна система, зворотний зв'язок, вектор керувань.

Новицкий В. В., Коломийчук О. П., Святовец И. Ф. МОДЕЛИРОВАНИЕ ПОЧТИ КОНСЕРВАТИВНОЙ СИСТЕМЫ С ПОМОЩЬЮ ОБРАТНОЙ СВЯЗИ / Ин-т математики НАН Украины, Запорожская государственная инженерная академия, Украина

Предлагается один из возможных подходов определения условий построения почти консервативной системы с помощью обратной связи. Приводятся примеры нахождения управления для построения почти консервативной системы.

Ключевые слова: почти консервативная система, обратная связь, вектор управления.

Novitsky V. V., Kolomiychuk O. P., Svyatovets I. F. MODELING THE ALMOST CONSERVATIVE SYSTEM USING FEEDBACK / In-t of Mathematics of NAS of Ukraine, Zaporizhzhya State Engineering Academy, Ukraine

This article examines the controlled linear stationary system even order. Matrix of coefficients of the variable has the form ($F_0+\varepsilon F_1$) where ε is a small parameter. The system is not almost conservative, i.e. condition of skew and nonsingularity is not satisfied. We solve the problem to formation almost conservative system