

# ДИФФЕРЕНЦИАЛЬНАЯ ГЕОМЕТРИЯ В МАХИМА

Учебное пособие

А.В. Букушева

Саратов  
2015

Букушева А.В. Дифференциальная геометрия в Mathematica. Учебное пособие. Саратов, 2015. 50 с.

Пособие предназначено для проведения лабораторных занятий дисциплины "Компьютерная геометрия и геометрическое моделирование". В учебном пособии приводятся примеры решения задач дифференциальной геометрии кривых и поверхностей в евклидовом пространстве в программе Mathematica. Учебное пособие адресовано студентам, обучающимся по направлению 02.03.01 - Математика и компьютерные науки. От читателя требуются знания основ дифференциальной геометрии и программирования. Пособие может оказаться полезным для читателей, интересующихся применением информационных технологий в геометрии.

## СОДЕРЖАНИЕ

1. Кривые на плоскости.....	5
1.1. Различные способы задания кривых.....	5
1.2. Длина дуги. Кривизна.....	10
1.3. Эволюты и эвольвенты. Натуральные уравнения.....	12
2. Кривые и поверхности в пространстве.....	17
Список литературы.....	42
Приложение.....	43

## Введение

В настоящее время распространёнными системами компьютерной математики являются Maple, Mathematica, MatLAB, MathCAD. Эти системы реализуют множество математических операций, снабжены мощными графическими средствами и обладают собственными языками программирования. Всё это предоставляет широкие возможности для применения математических пакетов в научных исследованиях и преподавании, в частности, для визуализации математических объектов и понятий. Однако все названные программы являются лицензионными. Для организации самостоятельной внеаудиторной работы студентов удобно использовать свободное программное обеспечение. Среди свободно распространяемых программ популярными являются Maxima, GeoGebra, Axiom, Scilab.

Основными преимуществами программы Maxima являются: возможность функционирования под управлением различных операционных систем, широкий класс решаемых задач, интерфейс программы на русском языке, наличие справки и инструкций по работе с программой. На официальном сайте Maxima, на многих образовательных порталах можно найти статьи и пособия с примерами использования программы при решении различных задач, например [3, 6].

Среди возможностей системы Maxima - большая библиотека функций для решения задач линейной алгебры, построение двумерных и трёхмерных графиков, использование широкого спектра математических функций, дифференцирование и интегрирование функций и многое другое.

В данной работе показаны некоторые возможности использования Maxima при решении задач дифференциальной геометрии кривых и поверхностей в евклидовом пространстве.

Пример использования Maxima в тензорных расчетах приведен в работе [1], [2]. В работе [3] показано применение Maxima для автоматизации символьных вычислений в документах системы LaTeX.

## 1. Кривые на плоскости

### 1.1. Различные способы задания кривых

1. Окружность радиуса  $r$  катится без скольжения по окружности радиуса  $R$ , оставаясь вне ее. Визуализировать траекторию точки  $M$  катящейся окружности (*эпициклоида*). Что будет при  $r = R$ ?

**Решение.** Поместим начало координат в центр неподвижной окружности. Будем считать, что в исходном положении точка  $M$  совпадает с точкой  $A$ , в которой катящаяся окружность касается неподвижной, а ось абсцисс направим через точку  $A$ . Обозначим:  $t = \widehat{MO_1N}$ ,  $m = r/R$ . Так как

$$\smile AN = \smile MN,$$

$$R \cdot \angle(NOA) = rt \Rightarrow \angle(NOA) = \frac{r}{R}t = mt.$$

Имеем

$$x = OP = OD + DP = OD + EM = (R + r)\cos mt + r \sin \widehat{MO_1E},$$

$$y = MP = O_1D - O_1E = (R + r)\sin mt - r \cos \widehat{MO_1E}.$$

Так как

$$\sin \widehat{MO_1E} = \sin(t - \widehat{OO_1D}) = \sin(t - (\frac{\pi}{2} - mt)) = -\cos(t + mt),$$

$$\cos \widehat{MO_1E} = \sin(t + mt), \quad r = mR,$$

то

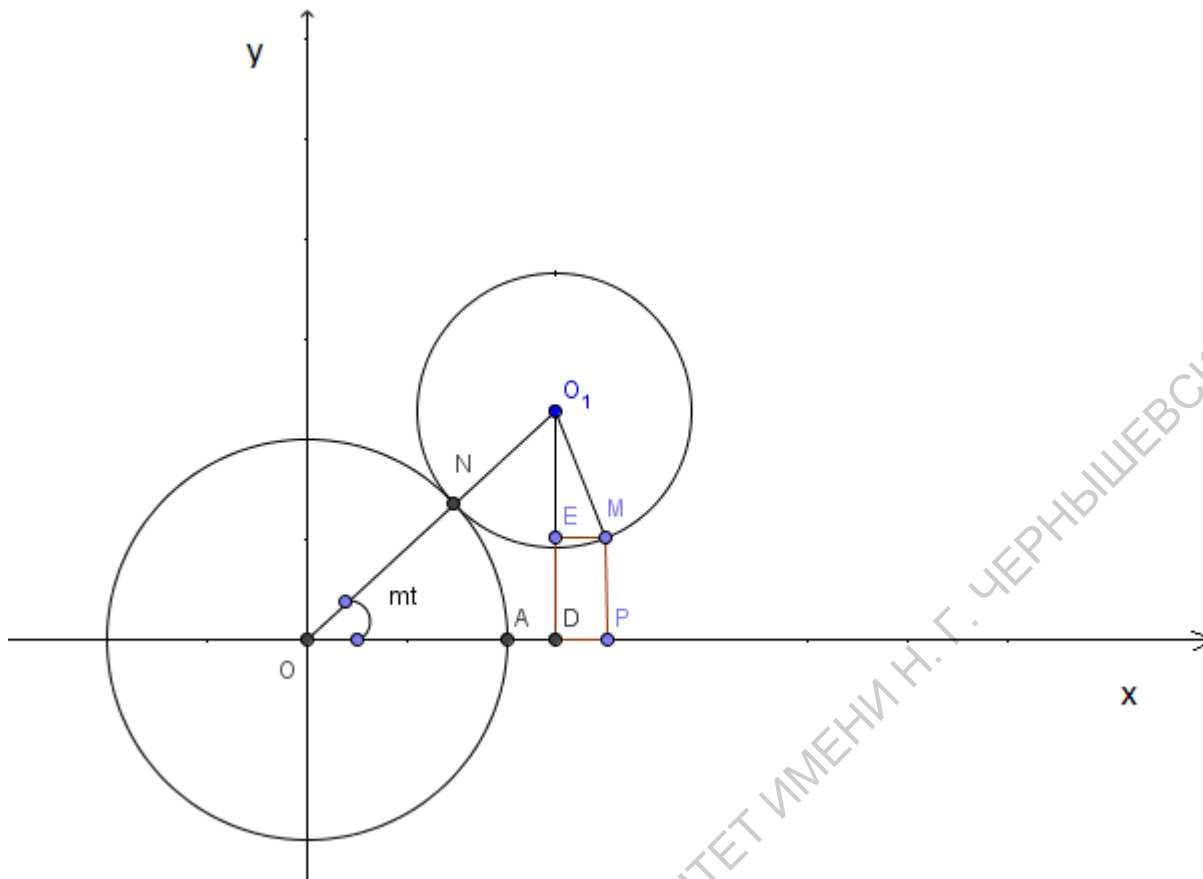
$$x = (R + mR)\cos mt - mR \cos(t + mt),$$

$$y = (R + mR)\sin mt - mR \sin(t + mt).$$

Исключая  $m$ , получаем

$$x = (R + r)\cos \frac{r}{R}t - r \cos \frac{R + r}{R}t,$$

$$y = (R + r)\sin \frac{r}{R}t - r \sin \frac{R + r}{R}t.$$



Все команды вводятся в поле ВВОД, разделителем команд является символ ; (точка с запятой). После ввода команды необходимо нажать клавиши [Shift] и [Enter] для ее обработки и вывода результата. Завершение ввода символом \$ (вместо точки с запятой) позволяет вычислить результат введенной команды, но не выводить его на экран.

После ввода каждой команде присваивается порядковый номер (%i1), (%i2) и т.д. Результаты вычислений имеют соответственно порядковый номер (%o1), (%o2) и т.д., где "i" сокращение от англ. input (ввод), а "o" англ. output (вывод). Это позволяет при дальнейшей записи команд сослаться на ранее записанные, например (%i1)+(%i2) будет означать добавление к выражению первой команды выражения второй с последующим вычислением результата.

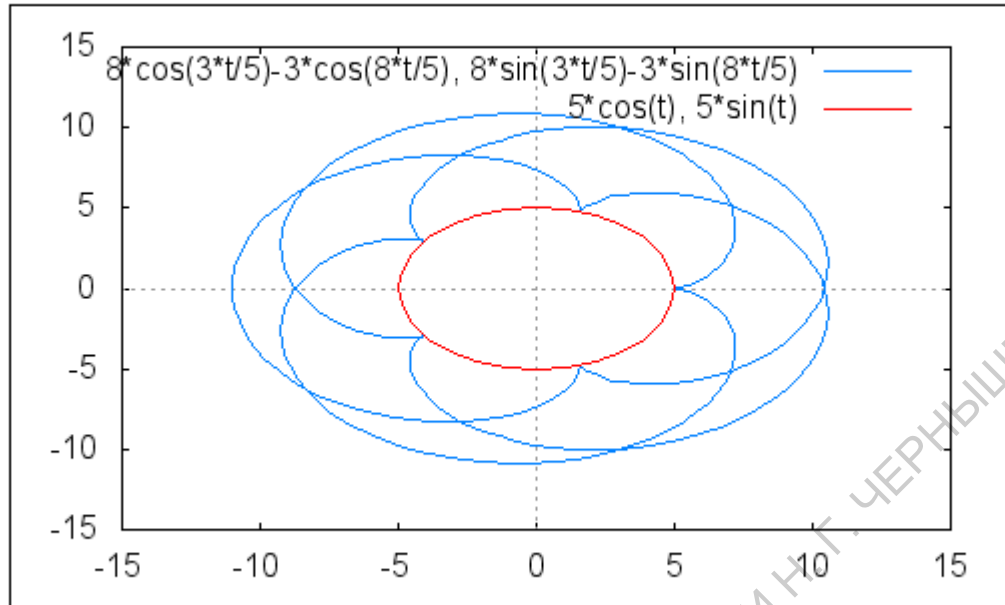
```
(%i1) c(t):=[(R+r)*cos(t*r/R)-r*cos(t*(R+r)/R),
            (R+r)*sin(t*r/R)-r*sin(t*(R+r)/R)];
```

```
(%o1) c(t):=[(R+r)cos(t*r/R)-r*cos(t*(R+r)/R),
            (R+r)sin(t*r/R)-r*sin(t*(R+r)/R)]
```

```
(%i2) R:5$
      r:3$
```

```
(%i4) wxplot2d([[parametric,c(t)[1],c(t)[2],[t,0,%pi*10],
[nticks,360]], [parametric,R*cos(t),R*sin(t),[t,0,%pi*2]]]);
```

(%t4)



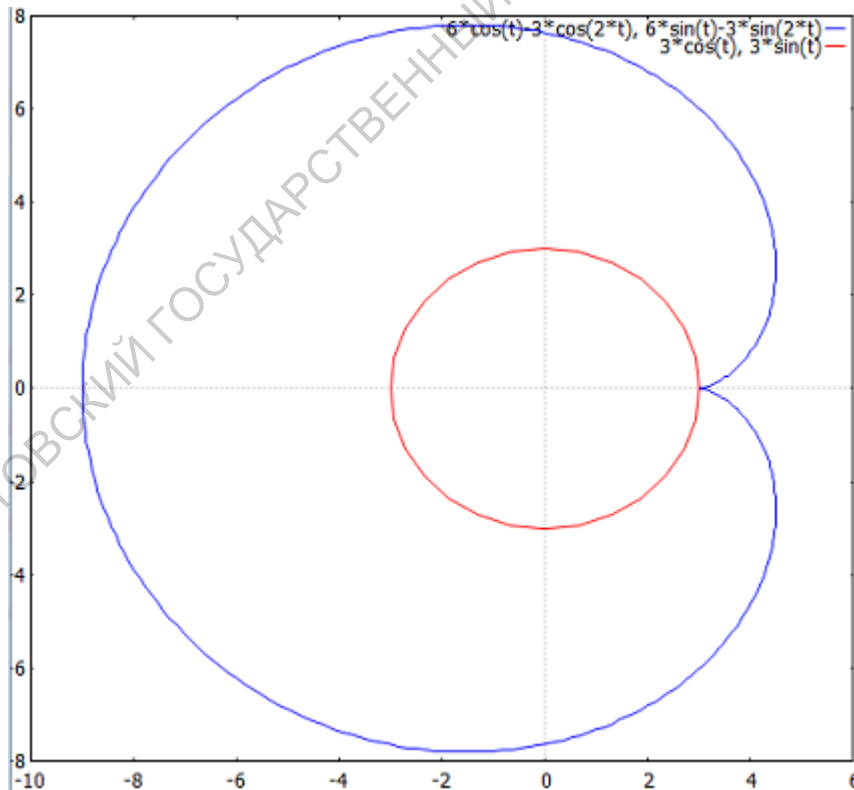
(%o4)

При  $r = R$  получаем кардиоиду.

```
(%i5) R:3$
r:3$
```

```
(%i7) plot2d([[parametric,c(t)[1],c(t)[2],[t,0,%pi*2],
[nticks,360]],
[parametric,R*cos(t),R*sin(t),[t,0,%pi*2]]]);
```

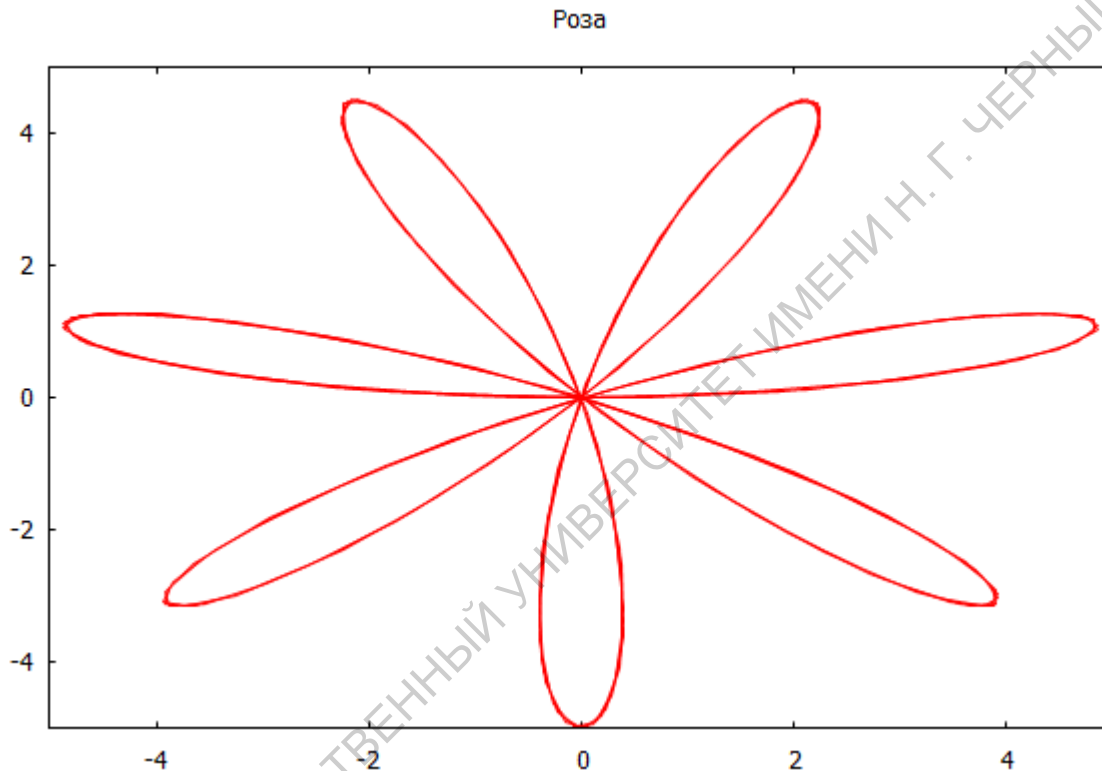
(%o7)



2. Построить кривую  $r = a \sin(b\varphi)$  в полярной системе координат

**Решение.**

```
(%i1) load(draw)$  
a:5$  
b:7$  
draw2d(nticks=300,  
        xrange=[-5,5],  
        yrange=[-5,5],  
        color=red,  
        title="Роза",  
        polar(a*sin(b*phi),phi,1,4*pi))$
```



3. Найти точки пересечения и углы, под которыми пересекаются линии:  
 $x^2 + y^2 = 9$ ,  $x^2 + y^2 - 6x = 9$ .

**Решение.**

```
(%i1) eq1:x^2+y^2=9$  
eq2:x^2+y^2-6*x=9$  
(%i3) res:solve([eq1,eq2],[x,y])  
/*координаты точек пересечения*/;  
(%o3) [[x=0,y=3],[x=0,y=-3]]  
(%i4) f(x,y):=x^2+y^2-9$  
g(x,y):=x^2+y^2-6*x-9$
```

Вычислим координаты касательных векторов:



```
(%i6) v:[diff(f(x,y),x),diff(f(x,y),y)];
(%o6) [2 x,2 y]

(%i7) w:[diff(g(x,y),x),diff(g(x,y),y)];
(%o7) [2 x-6,2 y]
```

Находим косинус угла между касательными векторами:

```
(%i8) v.w/(sqrt(v.v)*sqrt(w.w));
(%o8) 
$$\frac{4y^2+2x(2x-6)}{\sqrt{4y^2+4x^2}\sqrt{4y^2+(2x-6)^2}}$$

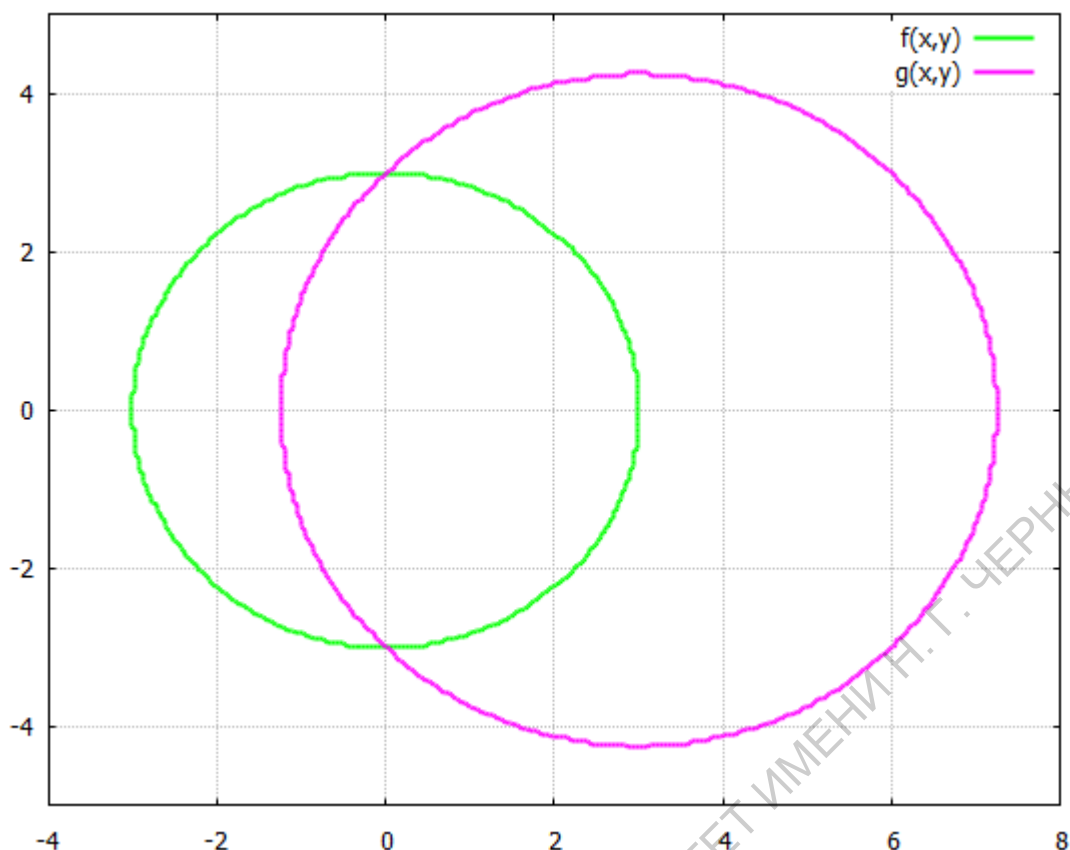
```

Вычислим косинус угла между касательными векторами в точках (0;3) и (0;-3):

```
(%i9) subst([res[1]],%);
      subst([res[2]],%);
(%o9)  $\frac{1}{\sqrt{2}}$ 
(%o10)  $\frac{1}{\sqrt{2}}$ 
(%i11) acos(%);
(%o11)  $\frac{\pi}{4}$ 
```

Данные линии пересекаются под углом  $\frac{\pi}{4}$ .

```
(%i12) load(draw)$
draw2d( nticks = 100,grid= true,
        line_width =2,
        key="f(x,y)",color=green,
        implicit(x^2+y^2-9, x, -4,8, y, -5,5),
        color=magenta,
        key= "g(x,y)",
        implicit(x^2+y^2-6*x=9, x,-4,8, y,-5,5))$
```



## 1.2. Длина дуги. Кривизна

4. Вычислите длину дуги между указанными точками кривой:  $x = 8at^3$ ,  $y = 3a(2t^2 - t^4)$ ,  $t_1 = 0$ ,  $t_2 = \sqrt{2}$ .

**Решение.** Длина дуги кривой, заданной уравнениями  $x = x(t)$ ,  $y = y(t)$ , вычисляется по формуле  $s = \int_{t_1}^{t_2} \sqrt{x'^2 + y'^2} dt$ .

```
(%i1) l(c,t):=integrate(sqrt(diff(c(t),t).diff(c(t),t)),t,t1,t2);
```

```
(%o1) l(c,t):=∫t1t2√diff(c(t),t).diff(c(t),t)dt
```

Вводим данную кривую и точки.

```
(%i2) c(t):=[8*a*t^3,3*a*(2*t^2-t^4)];
```

```
(%o2) c(t):=[8 a t3,3 a (2 t2-t4)]
```

```
(%i3) t1:0;
      t2:sqrt(2);
```

```
(%o3) 0
```

```
(%o4) √2
```

Вычисляем длину дуги кривой при условии  $a > 0$ .

```
(%i5) l(c,t);
Is a positive or negative?p;
(%o5) 24 a
```

5. Вычислите кривизну кривой  $x = x(t)$ ,  $y = y(t)$ :

**Решение.** Кривизна кривой, заданной уравнениями  $x = x(t)$ ,  $y = y(t)$ , вычисляется по формуле:  $k = \frac{|x'y'' - x''y'|}{(x'^2 + y'^2)^{3/2}}$ . Напишем программу:

```
(%i1) J(a):=[-a[2],a[1]]$
(%i2) k(c,t):=
diff(c(t),t,2).J(diff(c(t),t))/(diff(c(t),t).diff(c(t),t))^(3/2)$
```

Вычислим кривизну кривой  $y = \sin t$ :

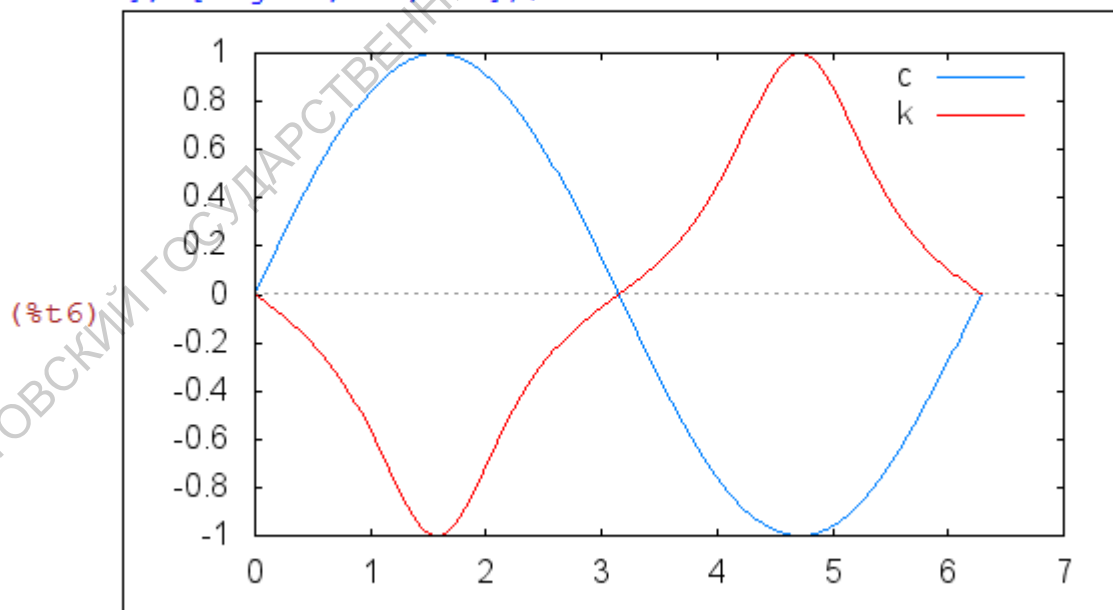
```
(%i3) c(t):=[t,sin(t)]$
(%i4) k(c,t)$ trigsimp(%);
```

```
(%o5) 
$$\frac{\sin(t)}{(\cos(t)^2 + 1)^{3/2}}$$

```

Изобразим кривые  $y = \sin t$  и  $y = -\frac{\sin t}{\sqrt{(\cos^2 t + 1)^3}}$ .

```
(%i6) wxplot2d([
[parametric,c(t)[1],c(t)[2],[t,0,%pi*2],[nticks,360]],
[parametric,t,k(c,t),[t,0,%pi*2],[nticks,360]]
],[legend,"c","k"]);
```



(%o6)

### 1.3. Эволюты и эвольвенты. Натуральные уравнения

6. Составьте уравнение эволюты кривой  $x = x(t)$ ,  $y = y(t)$  в Maxima. Составьте и нарисуйте эволюту кривой  $x = a \cos t$ ,  $y = b \sin t$ .

**Решение.** Эволюта, т.е. фигура состоящая из центров кривизны кривой, заданной уравнениями  $x = x(t)$ ,  $y = y(t)$ , имеет уравнения

$$X = x - y' \frac{x'^2 + y'^2}{x'y'' - x''y'}, \quad Y = y + x' \frac{x'^2 + y'^2}{x'y'' - x''y'}.$$

Зададим вектор:

```
(%i1) J(a):=[-a[2],a[1]];
(%o1) J(a):=[-a2,a1]
```

Вычислим длину касательного вектора кривой:

```
(%i2) v(c,t):=sqrt(diff(c(t),t).diff(c(t),t));
(%o2) v(c,t):=sqrt(diff(c(t),t).diff(c(t),t))
```

Найдем кривизну кривой  $x = x(t)$ ,  $y = y(t)$ :

```
(%i3) k(c,t):=
      diff(c(t),t,2).J(diff(c(t),t))/(diff(c(t),t).diff(c(t),t))^(3/2);
(%o3) k(c,t):=
      diff(c(t),t,2).J(diff(c(t),t))
      (diff(c(t),t).diff(c(t),t))3/2
```

Единичный вектор касательной кривой:

```
(%i4) T(c,t):=diff(c(t),t)/v(c,t);
(%o4) T(c,t):=
      diff(c(t),t)
      v(c,t)
```

Уравнение эволюты кривой  $x = x(t)$ ,  $y = y(t)$ :

```
(%i5) evoluta(c,t):=c(t)+1/k(c,t)*J(T(c,t));
(%o5) evoluta(c,t):=c(t)+
      1
      k(c,t) J(T(c,t))
```

Составим и нарисуем эволюту эллипса  $x = a \cos t$ ,  $y = b \sin t$  ( $a > 0$ ,  $b > 0$ ).

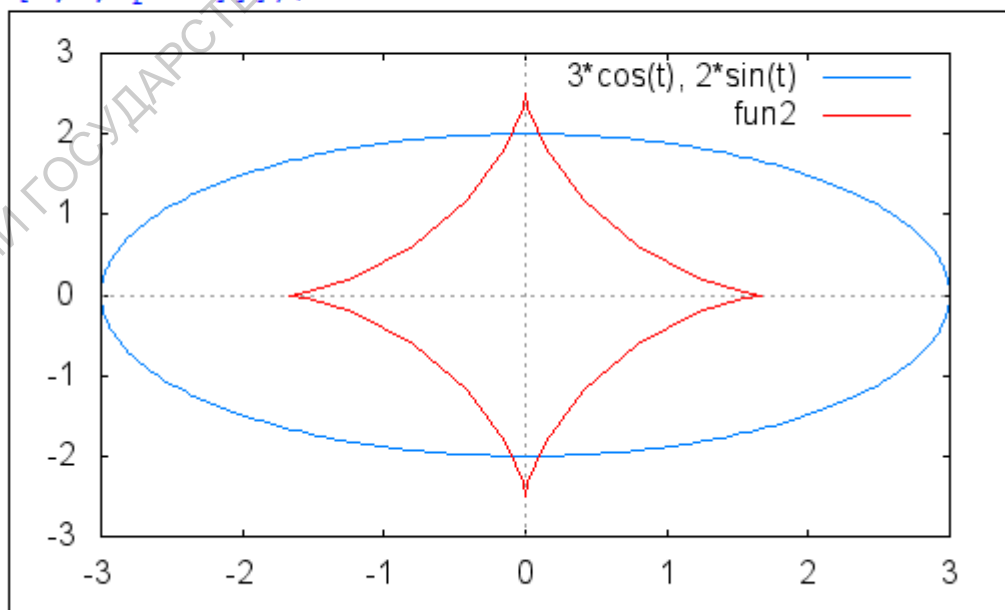
```
(%i6) assume(a>0,b>0);
(%o6) [a>0,b>0]

(%i7) c(t):=[a*cos(t),b*sin(t)];
(%o7) c(t):=[a*cos(t),b*sin(t)]
```

```
(%i8) trigsimp(v(c,t));
(%o8)  $\sqrt{a^2 \sin(t)^2 + b^2 \cos(t)^2}$ 
(%i9) trigsimp(T(c,t));
(%o9)  $\left[ -\frac{a \sin(t)}{\sqrt{a^2 \sin(t)^2 + b^2 \cos(t)^2}}, \frac{b \cos(t)}{\sqrt{a^2 \sin(t)^2 + b^2 \cos(t)^2}} \right]$ 
(%i10) k(c,t);
(%o10)  $\frac{a b \sin(t)^2 + a b \cos(t)^2}{(a^2 \sin(t)^2 + b^2 \cos(t)^2)^{3/2}}$ 
(%i11) trigsimp(k(c,t));
(%o11)  $\frac{a b}{(a^2 \sin(t)^2 + b^2 \cos(t)^2)^{3/2}}$ 
(%i12) trigsimp(evoluta(c,t));
(%o12)  $\left[ -\frac{(b^2 - a^2) \cos(t)^3}{a}, \frac{(b^2 - a^2) \sin(t)^3}{b} \right]$ 
```

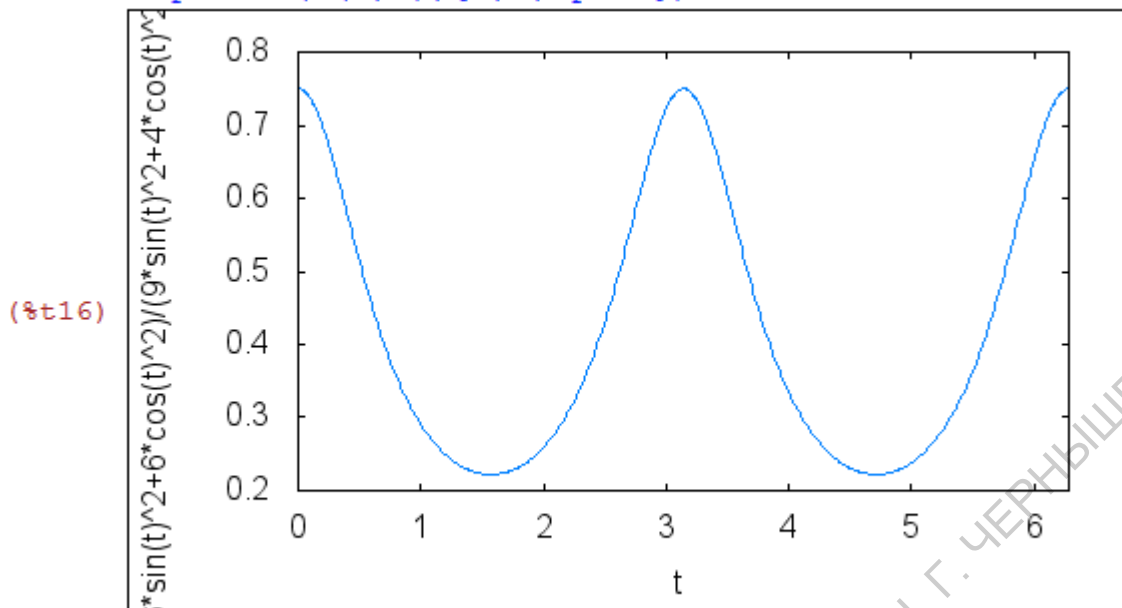
Уравнения эволюты эллипса имеют вид:  $X = -\frac{b^2-a^2}{a} \cos^3 t$ ,  $Y = \frac{b^2-a^2}{b} \sin^3 t$ .  
Кривизна эллипса равна  $k = \frac{ab}{(a^2 \sin^2 t + b^2 \cos^2 t)^{3/2}}$

```
(%i13) a:3$
      b:2$
(%i15) wxplot2d([[parametric,c(t)[1],c(t)[2],[t,0,%pi*2],
[nticks,360]],
[parametric,evoluta(c,t)[1],evoluta(c,t)[2],
[t,0,%pi*2]]]);
```



```
(%o15)
```

```
(%i16) wxplot2d(k(c,t), [t, 0, %pi*2]);
```



```
(%o16)
```

7. Составьте уравнения эвольвент окружности  $x^2 + y^2 = a^2$  и сделайте рисунок.

**Решение.** Эвольвентой данной кривой  $\gamma$  называется кривая  $\gamma^*$ , по отношению к которой  $\gamma$  является эволютой. Если кривая задана уравнением  $\vec{r} = \vec{r}(s)$ , то уравнение семейства ее эвольвент имеет вид  $\vec{\rho} = \vec{r} + (\lambda - s)\vec{t}$ , где  $\vec{t}$  - единичный вектор касательной к кривой  $\gamma$ , а  $\lambda$  - произвольный параметр.

Вводим параметрические уравнения окружности и вычисляем единичный касательный вектор.

```
(%i1) c:[a*cos(t),a*sin(t)];
```

```
(%o1) [a*cos(t),a*sin(t)]
```

```
(%i2) assume(a>0);
```

```
(%o2) [a>0]
```

```
(%i3) w:diff(c,t);
```

```
(%o3) [-a*sin(t),a*cos(t)]
```

```
(%i4) w/sqrt(w.w);
```

```
(%o4) [- $\frac{a \sin(t)}{\sqrt{a^2 \sin^2(t) + a^2 \cos^2(t)}}$ ,  $\frac{a \cos(t)}{\sqrt{a^2 \sin^2(t) + a^2 \cos^2(t)}}$ ]
```

```
(%i5) v:trigsimp(%);
```

```
(%o5) [-sin(t),cos(t)]
```

```
(%i6) q:c+(C-t)*v;
```

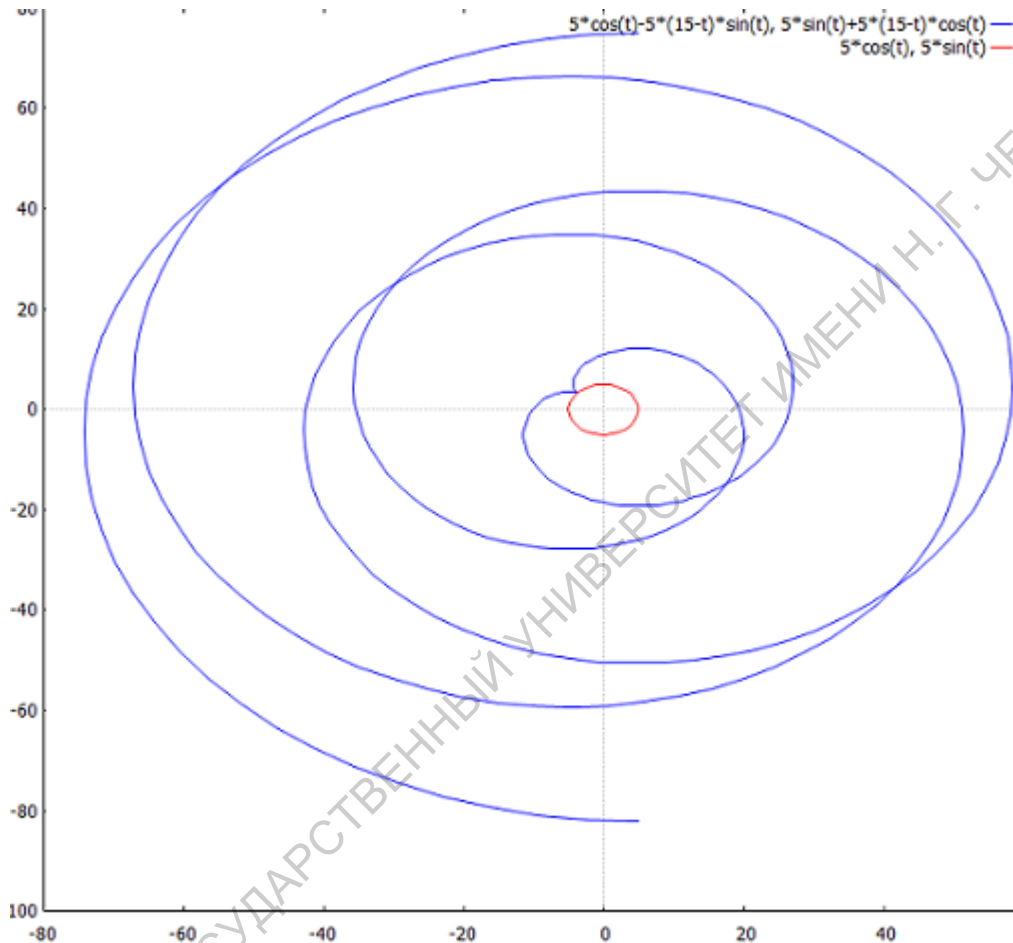
```
(%o6) [a*cos(t)-sin(t)*(C-t),cos(t)*(C-t)+a*sin(t)]
```

Уравнения эвольвент окружности имеют вид:  $X = a(\cos t + (t - C)\sin t)$ ,  
 $Y = a(\sin t - (t - C)\cos t)$ , где  $C$  - параметр семейства эвольвент.

```
(%i7) a:5$
      C:15$
```

```
(%i9) plot2d([[parametric,a*cos(t)-a*sin(t)*(C-t),
                a*cos(t)*(C-t)+a*sin(t),
                [t,0,%pi*10], [nticks,360]],
              [parametric,a*cos(t),a*sin(t),[t,0,%pi*2]]]);
```

```
(%o9)
```



8. Найти параметрические уравнения кривых, зная их натуральные уравнения:  $k = \frac{1}{s}$ .

**Решение.** Кривизна кривой вычисляется по формуле  $k = \frac{d\alpha}{ds}$ , где  $\alpha$  - угол, который образует касательная к кривой с осью  $Ox$ . Если заданы натуральные уравнения кривой, то параметризация кривой может быть задана в виде:  $x = \int \cos \alpha(s) ds$ ,  $y = \int \sin \alpha(s) ds$ .

```
(%i1) kappa:1/(a*s);
```

```
(%o1) 1
      a s
```

```
(%i2) theta:integrate(kappa,s);
```

```
(%o2) 
$$\frac{\log(s)}{a}$$

```

```
(%i3) X:integrate(cos(theta),s)$  
ratsimp(%);
```

```
(%o4) 
$$\frac{a s \sin\left(\frac{\log(s)}{a}\right) + a^2 s \cos\left(\frac{\log(s)}{a}\right)}{a^2 + 1}$$

```

```
(%i5) Y:integrate(sin(theta),s)$  
ratsimp(%);
```

```
(%o6) 
$$\frac{a^2 s \sin\left(\frac{\log(s)}{a}\right) - a s \cos\left(\frac{\log(s)}{a}\right)}{a^2 + 1}$$

```

```
(%i7) kill(all);
```

```
(%o0) done
```

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО



## 2. Кривые и поверхности в пространстве

9. Постройте кривую Вивиана:  $x = \frac{R}{2}(1 + \cos t)$ ,  $y = \frac{R}{2}\sin t$ ,  $z = \pm R \sin \frac{t}{2}$ . Это кривая, образ которой есть пересечение сферы радиуса  $R$  и кругового цилиндра диаметра  $R$ , одна из образующих которого проходит через центр сферы.

```
(%i1) R:2$
```

```
(%i2) curve:parametric(R/2*(1+cos(t)), R/2*sin(t),  
R*sin(t/2),t,0,4*pi);
```

```
(%o2) parametric(cos(t)+1, sin(t), 2 sin(t/2), t, 0, 4 pi)
```

Построим сферу и цилиндр, используя команду `implicit`, которая позволяет строить графики функций, заданных неявно.

```
(%i3) sphere:implicit(R^2=x^2+y^2+z^2,  
x,-2,2,y,-2,2,z,-2,2);
```

```
(%o3) implicit(4=z^2+y^2+x^2, x, -2, 2, y, -2, 2, z, -2, 2)
```

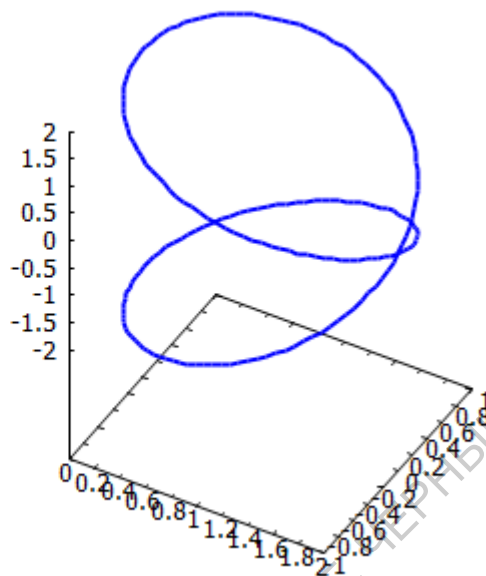
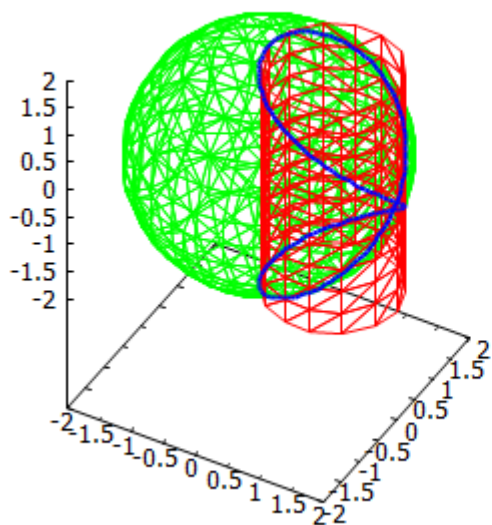
```
(%i4) cylinder:implicit(0=x^2+y^2-R*x,  
x,-2,2,y,-2,2,z,-2,2);
```

```
(%o4) implicit(0=y^2+x^2-2*x, x, -2, 2, y, -2, 2, z, -2, 2)
```

```
(%i5) scenel: gr3d(nticks=200,color=green,sphere,  
color=red,cylinder,  
line_width=2,color=blue,curve)$
```

```
(%i6) scene2: gr3d(nticks=200,  
line_width=2,color=blue,curve)$
```

```
(%i7) load(draw)$  
draw(scenel, scene2, columns = 2)$
```



10. Найти кривизну и кручение конической винтовой линии  $x = t \cos t$ ,  $y = t \sin t$ ,  $z = at$ . Раскрасить график линии в зависимости от её кривизны.

**Решение.** Загружаем пакет `load('vect')`. Кривизна кривой, заданной уравнениями  $x = x(t)$ ,  $y = y(t)$ ,  $z = z(t)$ , вычисляется по формуле  $k =$

$$\frac{\sqrt{\begin{vmatrix} y' & z' \\ y'' & z'' \end{vmatrix}^2 + \begin{vmatrix} z' & x' \\ z'' & x'' \end{vmatrix}^2 + \begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}^2}}{(x'^2 + y'^2 + z'^2)^{3/2}}$$

```
(%i1) load("vect")$
(%i2) n(c,t):=express(diff(c(t),t)~diff(c(t),t,2));
(%o2) n(c,t):=express(diff(c(t),t)~diff(c(t),t,2))

(%i3) k(c,t):=
sqrt(n(c(t),t) . n(c(t),t)) / (diff(c(t),t) . diff(c(t),t))^(3/2);
(%o3) k(c,t):=

$$\frac{\sqrt{n(c(t),t) \cdot n(c(t),t)}}{(diff(c(t),t) \cdot diff(c(t),t))^{3/2}}$$

```

Кручение кривой, заданной уравнениями  $x = x(t)$ ,  $y = y(t)$ ,  $z = z(t)$ ,

вычисляется по формуле  $\kappa = \frac{\begin{vmatrix} x' & y' & z' \\ x'' & y'' & z'' \\ x''' & y''' & z''' \end{vmatrix}}{\begin{vmatrix} y' & z' \\ y'' & z'' \end{vmatrix}^2 + \begin{vmatrix} z' & x' \\ z'' & x'' \end{vmatrix}^2 + \begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}^2}$

```
(%i4) v1(c,t):=diff(c(t),t)$
      v2(c,t):=diff(c(t),t,2)$
      v3(c,t):=diff(c(t),t,3)$
```

```
(%i7) I(c,t):=matrix([v1(c,t)[1],v1(c,t)[2],v1(c,t)[3]],
                    [v2(c,t)[1],v2(c,t)[2],v2(c,t)[3]],
                    [v3(c,t)[1],v3(c,t)[2],v3(c,t)[3]]);
```

```
(%o7) I(c,t):=
      [ (v1(c,t))_1 (v1(c,t))_2 (v1(c,t))_3
        (v2(c,t))_1 (v2(c,t))_2 (v2(c,t))_3
        (v3(c,t))_1 (v3(c,t))_2 (v3(c,t))_3 ]
```

```
(%i8) kappa(c,t):=determinant(I(c,t))/(n(c,t).n(c,t))^2;
```

```
(%o8) kappa(c,t):=
      determinant(I(c,t))
      (n(c,t).n(c,t))^2
```

Вычислим кривизну и кручение кривой конической винтовой линии при  $a > 0$ .

```
(%i9) assume(a>0);
(%o9) [a>0]
```

```
(%i10) c(t):=[t*cos(t),t*sin(t),a*t];
(%o10) c(t):=[t*cos(t),t*sin(t),a*t]
```

```
(%i11) trigsimp(kappa(c,t));
```

```
(%o11) 
$$\frac{\sqrt{t^4+(a^2+4)t^2+4a^2+4}}{(t^2+a^2+1)^{3/2}}$$

```

```
(%i12) trigsimp(kappa(c,t));
```

```
(%o12) 
$$\frac{a t^2 + 6 a}{t^8 + (2 a^2 + 8) t^6 + (a^4 + 16 a^2 + 24) t^4 + (8 a^4 + 40 a^2 + 32) t^2 + 16 a^4 + 32 a^2 + 16}$$

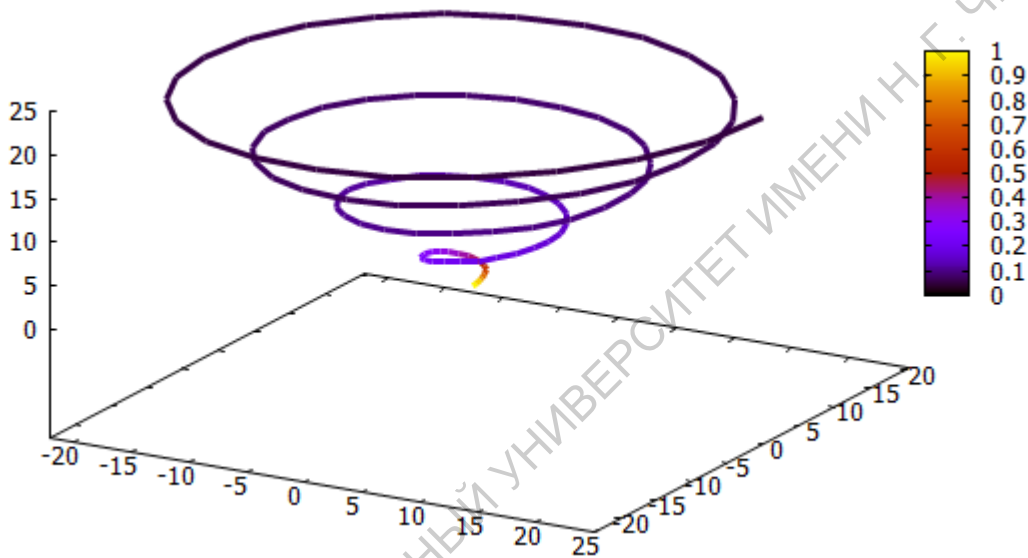
```

Вычислим кривизну (команда subst) и изобразим винтовую линию при  $a = 1$ .

```
(%i13) a:1;
(%o13) 1

(%i14) subst([a=1],sqrt(t^4+(a^2+4)*t^2+4*a^2+4)/(t^2+a^2+1)^(3/2));
(%o14) 
$$\frac{\sqrt{t^4+5t^2+8}}{(t^2+2)^{3/2}}$$


(%i15) load(draw)$
draw3d(
nticks = 100,
line_width = 3,
enhanced3d = sqrt(t^4+5*t^2+8)/(t^2+2)^(3/2),
parametric(t*cos(t),t*sin(t),a*t,t,0,8*%pi))$
```



11. Вычислим векторы  $\vec{r}_u = \frac{\partial \vec{r}}{\partial u}$ ,  $\vec{r}_v = \frac{\partial \vec{r}}{\partial v}$ ,  $\vec{n} = \frac{[\vec{r}_u \vec{r}_v]}{||[\vec{r}_u \vec{r}_v]||}$ ,  $\vec{n}_u = \frac{\partial \vec{n}}{\partial u}$ ,  $\vec{n}_v = \frac{\partial \vec{n}}{\partial v}$  в точке  $(0,0)$  поверхности  $\vec{r}(u,v) = (u,v,u^2 - v^2)$ .

```
(%i1) load("vect")$
(%i2) r:[u,v,u^2-v^2];
(%o2) [u,v,u^2-v^2]
(%i3) ru:diff(r,u); subst([u=0,v=0],%);
(%o3) [1,0,2u]
(%o4) [1,0,0]
```

```

(%i5) rv:diff(r,v); subst([u=0,v=0],%);
(%o5) [0,1,-2 v]
(%o6) [0,1,0]

(%i7) nn:express(ru~rv);
(%o7) [-2 u,2 v,1]

(%i8) n:nn/sqrt(nn.nn);
(%o8) [-\frac{2 u}{\sqrt{4 v^2+4 u^2+1}},\frac{2 v}{\sqrt{4 v^2+4 u^2+1}},\frac{1}{\sqrt{4 v^2+4 u^2+1}}]

(%i9) diff(n,u);nu:subst([u=0,v=0],%);
(%o9) [-\frac{8 u^2}{(4 v^2+4 u^2+1)^{3/2}}-\frac{2}{\sqrt{4 v^2+4 u^2+1}},-\frac{8 u v}{(4 v^2+4 u^2+1)^{3/2}},-\frac{4 u}{(4 v^2+4 u^2+1)^{3/2}}]
(%o10) [-2,0,0]

(%i11) diff(n,v);nv:subst([u=0,v=0],%);
(%o11) [-\frac{8 u v}{(4 v^2+4 u^2+1)^{3/2}},\frac{2}{\sqrt{4 v^2+4 u^2+1}},-\frac{8 v^2}{(4 v^2+4 u^2+1)^{3/2}},-\frac{4 v}{(4 v^2+4 u^2+1)^{3/2}}]
(%o12) [0,2,0]

(%i13) kill(all);
(%o0) done

```

12. Вычислите полную и среднюю кривизны прямого геликоида  $x = u \cos v$ ,  $y = u \sin v$ ,  $z = av$ .

**Решение.** Изобразим геликоид.

```

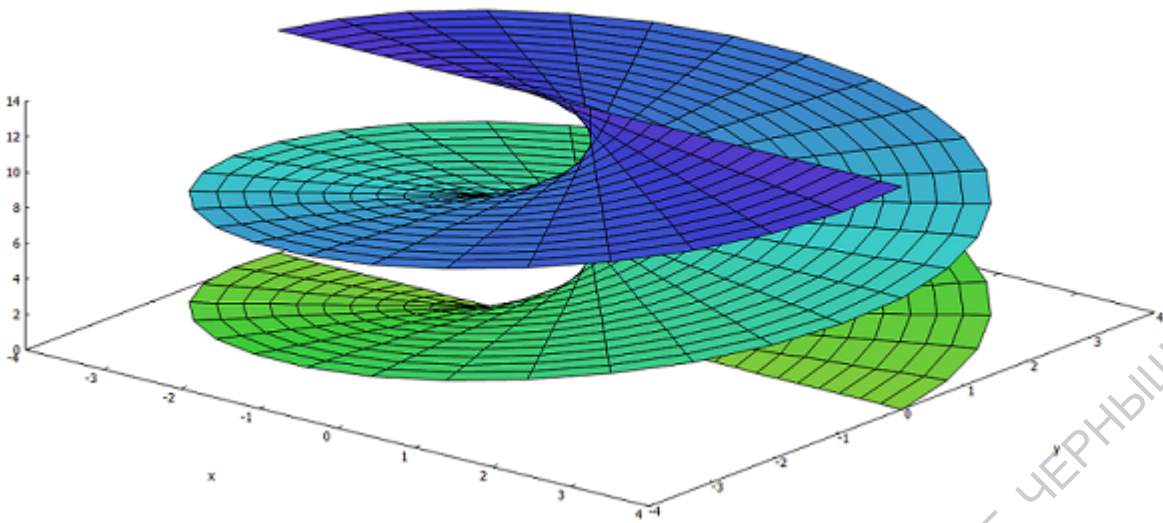
(%i1) r(u,v):=[u*cos(v), u*sin(v), a*v];
(%o1) r(u,v):=[u cos(v), u sin(v), a v]

(%i2) a:2$

(%i3) plot3d(r(u,v), [v,0,2*%pi], [u,-4,4],
[legend,"геликоид"]);$

(%i4) kill(all);
(%o0) done

```



```
(%i4) kill(all);
(%o0) done
```

Полной кривизной в точке поверхности называется произведение главных кривизн в этой точке:  $K = k_1 k_2$ . Средней кривизной в точке поверхности называется полусумма главных кривизн в этой точке:  $H = \frac{k_1 + k_2}{2}$ . Главные кривизны являются корнями уравнения  $\det(\Pi - \lambda I) = 0$ , где  $I(u, v) = \begin{pmatrix} (\vec{r}_u, \vec{r}_u) & (\vec{r}_u, \vec{r}_v) \\ (\vec{r}_v, \vec{r}_u) & (\vec{r}_v, \vec{r}_v) \end{pmatrix}$ ,  $\Pi(u, v) = \begin{pmatrix} (\vec{r}_{uu}, \vec{n}) & (\vec{r}_{uv}, \vec{n}) \\ (\vec{r}_{vu}, \vec{n}) & (\vec{r}_{vv}, \vec{n}) \end{pmatrix}$ ,  $\vec{n} = \frac{[\vec{r}_u, \vec{r}_v]}{||[\vec{r}_u, \vec{r}_v]||}$ .

```
(%i1) load("vect")$
(%i2) r(u,v):=[u*cos(v),u*sin(v),a*v];
(%o2) r(u,v):=[u*cos(v),u*sin(v),a*v]
(%i3) ru:=diff(r(u,v),u);
(%o3) [cos(v),sin(v),0]
(%i4) rv:=diff(r(u,v),v);
(%o4) [-u*sin(v),u*cos(v),a]
(%i5) n:=trigsimp(express(ru~rv));
(%o5) [a*sin(v),-a*cos(v),u]
(%i6) E:=trigsimp(ru.ru);
      F:=ru.rv;
      G:=trigsimp(rv.rv);
(%o6) 1
(%o7) 0
(%o8) u^2+a^2
```

```

(%i9) ruu:diff(ru,u);
(%o9) [0,0,0]

(%i10) ruv:diff(ru,v);
(%o10) [-sin(v),cos(v),0]

(%i11) rvv:diff(rv,v);
(%o11) [-u*cos(v),-u*sin(v),0]

(%i12) n0:trigsimp(n/sqrt(n.n));
(%o12) [frac(a*sin(v),sqrt(u^2+a^2)),frac(a*cos(v),sqrt(u^2+a^2)),frac(u,sqrt(u^2+a^2))]

(%i13) L:ruu.n0;
M:trigsimp(ruv.n0);
N:rvv.n0;
(%o13) 0

(%o14) frac(a,sqrt(u^2+a^2))

(%o15) 0

(%i16) I:matrix([E,F],[F,G]);
d:determinant(I);Ii:matrix([G,-F],[-F,E])/d;
(%o16) [1 0]
[0 u^2+a^2]

(%o17) u^2+a^2

(%o18) [1 0]
[0 1]
[u^2+a^2]

(%i19) W:Ii.matrix([L,M],[M,N]);
(%o19) [0,frac(a,sqrt(u^2+a^2))]
[frac(a,(u^2+a^2)^(3/2)),0]

(%i20) lambda:eigenvalues(W);
(%o20) [[frac(-a,u^2+a^2),frac(a,u^2+a^2)],[1,1]]

```

```
(%i21) K: lambda [1] [1] * lambda [1] [2];
```

```
(%o21) 
$$-\frac{a^2}{(u^2+a^2)^2}$$

```

```
(%i22) H: (lambda [1] [1] + lambda [1] [2]) / 2;
```

```
(%o22) 0
```

Ответ. Полная кривизна прямого геликоида равна  $K = -\frac{a^2}{(u^2+a^2)^2}$ , средняя кривизна -  $H = 0$ .

**13.** Найти уравнения сферических локсодром, т.е. кривых всюду имеющих один и тот же угол по отношению к меридианам. Визуализировать сферическую локсодрому, выходящую из данной точки в данном направлении.

**Решение.**

Параметрические уравнения сферы радиуса  $R$  имеют вид:

$$\vec{r} = R(\cos u \cos v, \sin u \cos v, \sin v).$$

Пусть искомая кривая  $\Gamma$  задана задана параметрическими уравнениями:

$$\Gamma : u = u(t), \quad v = v(t).$$

Меридиан задается уравнением:

$$\bar{\Gamma} : u = u_0.$$

Пусть эти кривые пересекаются в некоторой точке  $M_0(u_0, v_0)$ . Касательный вектор к меридиану в этой точке равен:

$$\vec{T}_m = \vec{r}_v|_{M_0} = R(-\cos u_0 \sin v_0, -\sin u_0 \sin v_0, \cos v_0).$$

Нормируем этот вектор:

$$\vec{\tau}_m = (-\cos u_0 \sin v_0, -\sin u_0 \sin v_0, \cos v_0).$$

Касательный вектор к искомой кривой равен:

$$\vec{T} = \vec{r}_t = R(-\sin u \cos v \dot{u} - \cos u \sin v \dot{v}, \cos u \cos v \dot{u} - \sin u \sin v \dot{v}, \cos v \dot{v})_{M_0}.$$

Длина этого вектора равна:

$$|\vec{T}| = R\sqrt{\dot{v}^2 + \cos^2 v \dot{u}^2}.$$



Полагая  $t = \theta$  и избавляясь от несущественного множителя, найдем:

$$\vec{T} = (-\sin u \cos v u' - \cos u \sin v, \cos u \cos v u' - \sin u \sin v, \cos v)_{M_0}$$

и

$$|\vec{T}| = \sqrt{1 + u'^2 \cos^2 v},$$

где обозначено  $u' = \frac{du}{dv}$ . Если  $\theta$  - угол между искомой линией и меридианом, то:

$$\cos \theta = \frac{(\vec{T}, \vec{\tau}_m)}{|\vec{T}| |\vec{\tau}|} = \frac{1}{\sqrt{1 + u'^2 \cos^2 v}}.$$

Возводим обе части равенства в квадрат и преобразуем :

$$\cos^2 \theta (1 + u'^2 \cos^2 v) = 1 \Rightarrow u'^2 \cos^2 v = \operatorname{tg}^2 \theta \Rightarrow \frac{dv}{\cos v} = \frac{du}{\operatorname{tg} \theta}.$$

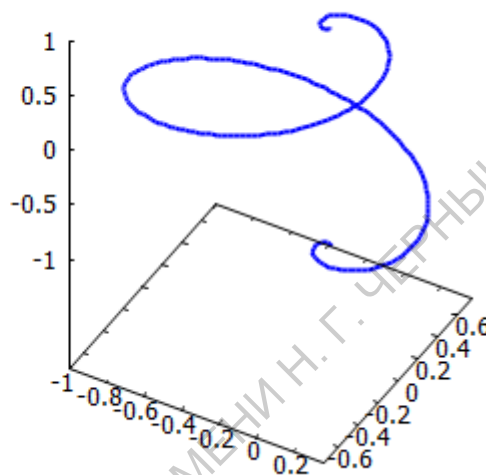
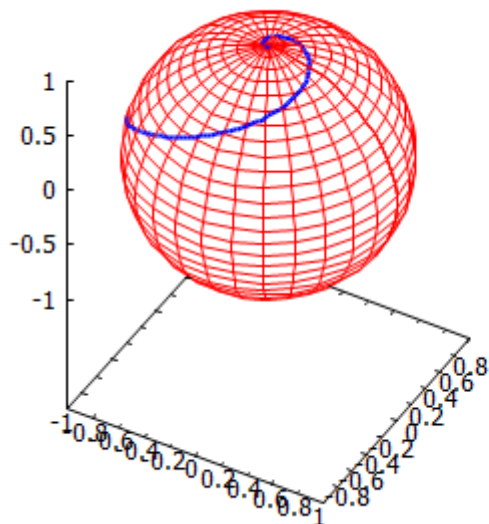
Интегрируя это дифференциальное уравнение первого порядка с разделяющимися переменными, найдем уравнение локсодромы.

Уравнение сферической локсодромы имеет вид:  $u = \operatorname{tg} \theta \ln (\operatorname{tg}(\pi/4 + v/2))$ .

```
(%i1) sphere:parametric_surface(
      cos(v)*cos(u),
      cos(v)*sin(u),
      sin(v),v,-%pi/2,%pi/2,u,0,2*pi);
(%o1) parametric_surface
      (cos(u)cos(v),sin(u)cos(v),sin(v),v,-pi/2,pi/2,u,0,2pi)
(%i2) curve:parametric(
      cos(t)*cos(a+tan(theta)*log(tan(pi/4+t/2))),
      cos(t)*sin(a+tan(theta)*log(tan(pi/4+t/2))),sin(t),
      t,-%pi/2,%pi/2);
(%o2) parametric(cos(t)cos(log(tan(t/2+pi/4))tan(theta)+a),cos(t)
      sin(log(tan(t/2+pi/4))tan(theta)+a),sin(t),t,-pi/2,pi/2)
(%i3) theta:%pi/3$
      a:%pi$
(%i5) scene1:gr3d(nticks=200,surface_hide=true,
      color=red,sphere,
      line_width=2,color=blue,curve)$
```

```
(%i6) scene2: gr3d(nticks=200,
line_width=2,color=blue,curve)$

(%i7) load(draw)$
draw(scenel, scene2, columns = 2)$
```



14. Визуализировать локсодрому поверхности:  $x = u \cos v, y = u \sin v, z = \ln(u + \sqrt{u^2 - 1})$ , выходящую из данной точки в данном направлении.

**Решение.** Составим уравнение локсодромы. Задаем параллель  $v$ , полагая  $v = v_0$ . Параметрическое задание параллели имеет вид:  $f(t) = (t, v_0)$ , тогда касательный вектор  $f'(t) = (1, 0)$ .

Пусть искомая кривая задается уравнениями:  $\alpha(t) = (\sigma(t), t)$ , тогда  $\alpha'(t) = (\sigma'(t), 1)$ .

Пусть касательные к параллели и локсодроме пересекаются под углом  $\phi$ . Тогда

$$\cos \phi = \frac{I(\alpha'(t), f'(t))}{|\alpha'(t)| |f'(t)|},$$

где

$$I(f'(t), \alpha'(t)) = (\sigma'(t), 1) \begin{pmatrix} E & F \\ F & G \end{pmatrix} (1, 0),$$

$$|\alpha'(t)| = \sqrt{I(\alpha'(t), \alpha'(t))},$$

решая это уравнение находим уравнение локсодромы.

Вводим параметрические уравнения данной поверхности.

```
(%i1) sphere: parametric_surface(
    u*cos(v),
    u*sin(v),
    log(u+sqrt(u^2-1)), u, 0, 7, v, 0, 2*%pi);
(%o1)
parametric_surface(u*cos(v), u*sin(v), log(sqrt(u^2-1)+u), u, 0, 7, v, 0, 2*pi)
```

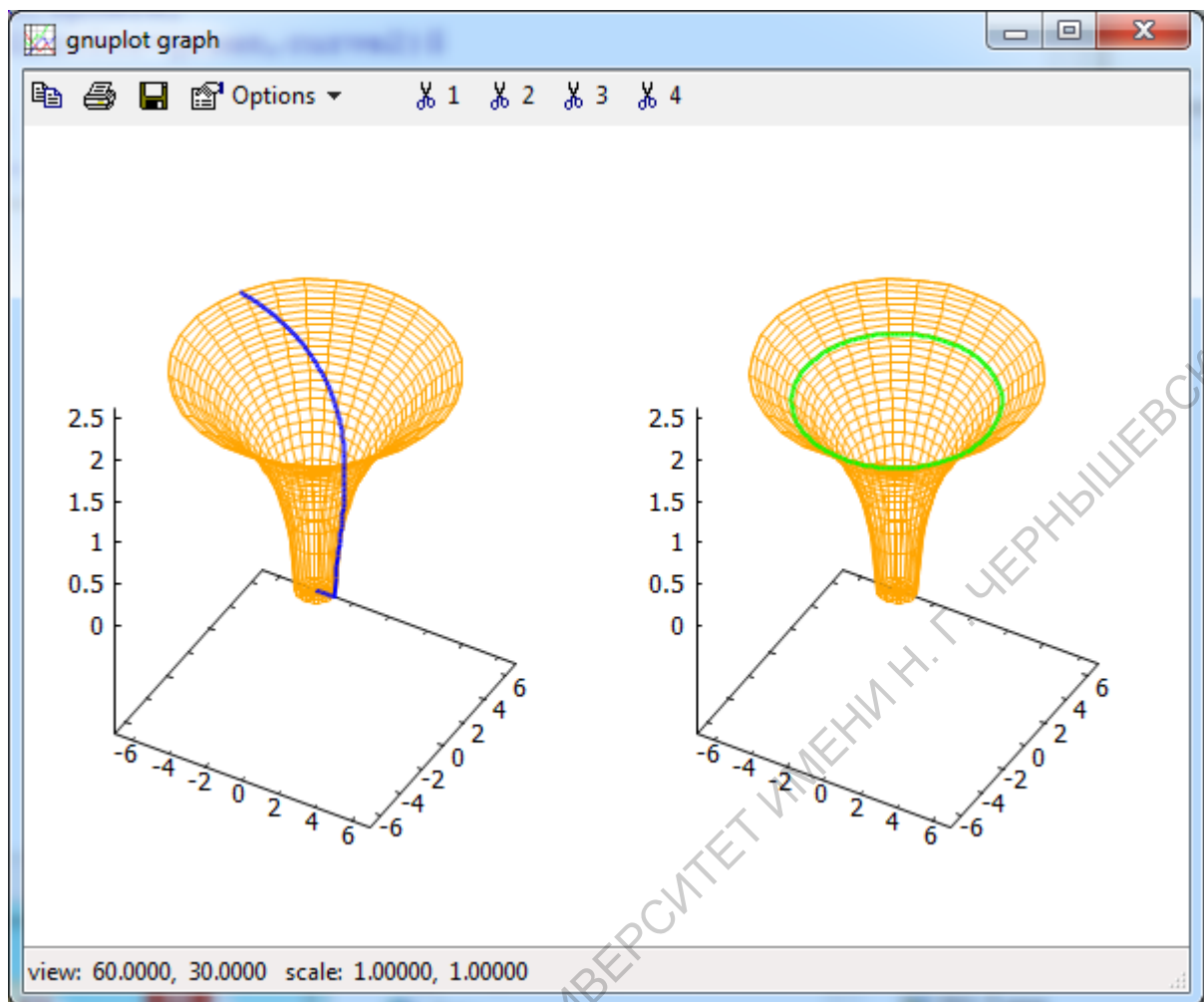
Задаем уравнение одной локсодромы при  $v = const$ .

```
(%i2) curve1: parametric(
    u*cos(log(u+sqrt(u^2-1))),
    u*sin(log(u+sqrt(u^2-1))), log(u+sqrt(u^2-1)),
    u, 0, 7);
(%o2) parametric
(u*cos(log(sqrt(u^2-1)+u)), u*sin(log(sqrt(u^2-1)+u)), log(sqrt(u^2-1)+u), u, 0, 7)
```

```
(%i3) scene1: gr3d( nticks=200,
    color=orange, sphere,
    line_width=2, color=blue, curve1) $
```

Задаем уравнение второй локсодромы при  $u = const$  ( $u=5$ )

```
(%i4) u: 5 $
(%i5) curve2: parametric(
    u*cos(v),
    u*sin(v), log(u+sqrt(u^2-1)),
    v, 0, 2*%pi);
(%o5) parametric(5*cos(v), 5*sin(v), log(2*sqrt(6)+5), v, 0, 2*pi)
(%i6) scene2: gr3d( nticks=200,
    color=orange, sphere,
    line_width=2, color=green, curve2) $
(%i7) load(draw) $
draw(scene1, scene2, columns = 2) $
```



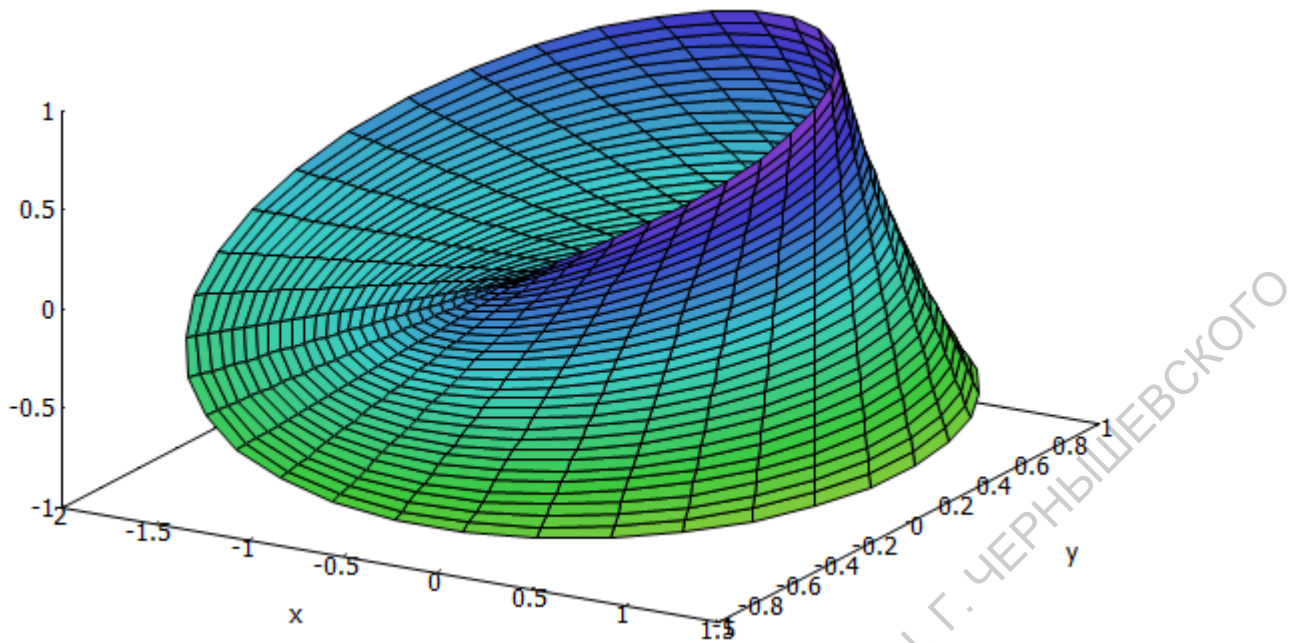
15. Изобразить ленту Мебиуса.

```
(%i1) alpha:[cos(u),sin(u),0];
(%o1) [cos(u),sin(u),0]

(%i2) E:[-sin(u/2),0,cos(u/2)];
(%o2) [-sin(u/2),0,cos(u/2)]

(%i3) r:alpha+v*E;
(%o3) [cos(u)-sin(u/2)v,sin(u),cos(u/2)v]

(%i5) plot3d(r,[u,0,2*pi],[v,-1,1])$
```



16. Изобразим псевдосферу и вычислим гауссову кривизну.

**Решение.** Пусть трактриса задана параметрическими уравнениями в плоскости  $xOy$ .

```
(%i1) load("vect")$
```

```
(%i2) c:[sin(u),cos(u)+log(tan(u/2)),0];assume(cos(u)>0);
```

```
(%o2) [sin(u),log(tan(u/2))+cos(u),0]
```

```
(%o3) [cos(u)>0]
```

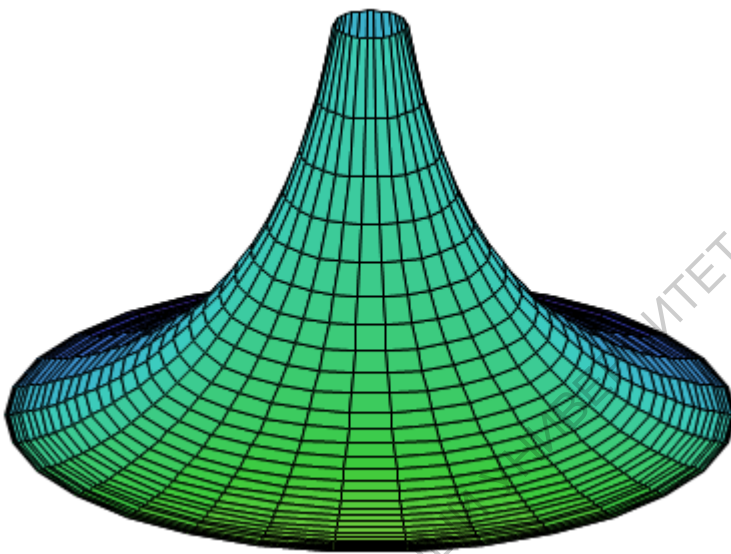
```
(%i4) R:matrix([cos(v),0,-sin(v)],[0,1,0],[sin(v),0,cos(v)]);
```

```
(%o4) [cos(v) 0 -sin(v)
       0 1 0
       sin(v) 0 cos(v)]
```

Получим параметрические уравнения псевдосферы.

```
(%i5) F:R.c;
(%o5) 
$$\begin{bmatrix} \sin(u) \cos(v) \\ \log\left(\tan\left(\frac{u}{2}\right)\right) + \cos(u) \\ \sin(u) \sin(v) \end{bmatrix}$$

(%i6) r:[F[1,1],F[2,1],F[3,1]];
(%o6)  $[\sin(u) \cos(v), \log\left(\tan\left(\frac{u}{2}\right)\right) + \cos(u), \sin(u) \sin(v)]$ 
plot3d(r, [u,0.1,1.5], [v,0,2*pi], [elevation,145], [azimuth,0],
[box,false], [legend,false])$
```



Вычислим полную и среднюю кривизну псевдосферы.

```
(%i8) ru:diff(r,u)$
(%i9) rv:diff(r,v)$
(%i10) express(ru~rv)$
(%i11) n:trigsimp(express(ru~rv))$
(%i12) n0:n/sqrt(factor(trigsimp(n.n)))$
(%i13) E:trigreduce(trigsimp(ru.ru))$
F:ru.rv$
G:factor(trigsimp(rv.rv))$
```

```

(%i16) ruu:diff(ru,u)$
(%i17) ruv:diff(ru,v)$
(%i18) rvv:diff(rv,v)$

(%i19) L:trigreduce(trigsimp(ruu.n0))$
      M:trigsimp(ruv.n0)$
      N:trigreduce(trigsimp(rvv.n0))$

I:matrix([E,F],[F,G]); d:determinant(I);Ii:matrix([G,-F],[-F,E])/d;

(%o22) 
$$\begin{bmatrix} -\csc(u)^2 \left( -\frac{\cos(2u)}{2} - \frac{1}{2} \right) & 0 \\ 0 & \sin(u)^2 \end{bmatrix}$$


(%o23) 
$$-\csc(u)^2 \sin(u)^2 \left( -\frac{\cos(2u)}{2} - \frac{1}{2} \right)$$


(%o24) 
$$\begin{bmatrix} \frac{1}{\csc(u)^2 \left( -\frac{\cos(2u)}{2} - \frac{1}{2} \right)} & 0 \\ 0 & \frac{1}{\sin(u)^2} \end{bmatrix}$$


(%i25) W:trigsimp(Ii.matrix([L,M],[M,N]))$
(%i26) l:eigenvalues(W)$
(%i27) G:trigreduce(l[1][1]*l[1][2]);
      M:trigreduce((l[1][1]+l[1][2])/2)$
(%o27) -1

```

Гауссова кривизна равна -1, средняя кривизна на печать не выведена.

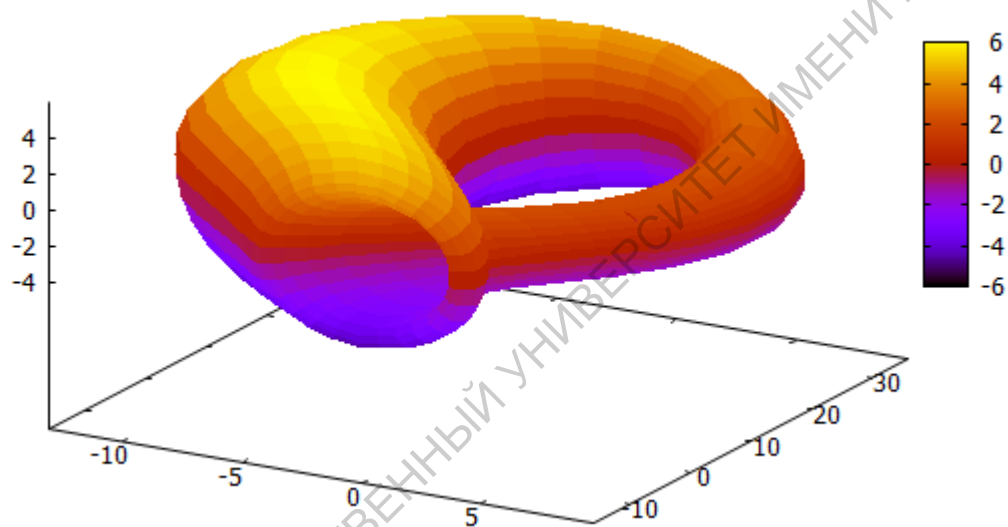
17. Изобразить бутылку Клейна.

```
(%i1) x1:6*cos(u)*(1+sin(u))+4*(1-0.5*cos(u))*cos(u)*cos(v)$
      x2:6*cos(u)*(1+sin(u))-4*(1-0.5*cos(u))*cos(u)*cos(v)$
      y1:16*sin(u)*(1+sin(u))+4*(1-0.5*cos(u))*sin(u)*cos(v)$
      y2:16*sin(u)$
      z1:4*(1-0.5*cos(u))*sin(v)$

(%i6) x:if u<%pi then x1 else x2$
      y:if u<%pi then y1 else y2$
      z:z1$

(%i9) load(draw)$
      draw3d(enhanced3d=true,
            parametric_surface(x,y,z,u,0,2*%pi,v,0,2*%pi));

(%o10) [gr3d(parametric_surface)]
```



18. Визуализировать геодезические на торе:  $\vec{r} = ((a + b\sin v)\cos u, (a + b\sin v)\sin u, b\cos v)$ .

Решение.

```
(%i1) load("vect")$

(%i2) r(u,v):=[(a+b*cos(u))*cos(v),
              (a+b*cos(u))*sin(v),
              b*sin(u)]$

(%i3) ru:diff(r(u,v),u)$

(%i4) rv:diff(r(u,v),v)$
```



```
(%i5) E:trigsimp(ru.ru)$
      F:ru.rv$
      G:trigsimp(rv.rv)$
```

```
(%i8) g:matrix([E,F],[F,G]);
```

```
(%o8) 
$$\begin{bmatrix} b^2 & 0 \\ 0 & b^2 \cos(u)^2 + 2 a b \cos(u) + a^2 \end{bmatrix}$$

```

В восьмой строке найдена матрица  $g_{ij}$ . Через G обозначена обратная матрица:  $G = (g^{ij})$ .

```
(%i9) G:invert(g);
```

```
(%o9) 
$$\begin{bmatrix} \frac{1}{b^2} & 0 \\ 0 & \frac{1}{b^2 \cos(u)^2 + 2 a b \cos(u) + a^2} \end{bmatrix}$$

```

Пользуясь формулой  $\Gamma_{ij}^k = \frac{1}{2}g^{kl} \left( \frac{\partial g_{lj}}{\partial x^i} + \frac{\partial g_{il}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^l} \right)$ , найдем символы Кристоффеля  $\Gamma_{12}^2, \Gamma_{22}^1, \Gamma_{12}^2$ , остальные коэффициенты равны нулю.

```
(%i10) g212:trigsimp(1/2*G[2,2]*diff(g[2,2],u));
```

```
(%o10) 
$$\frac{b \sin(u)}{b \cos(u) + a}$$

```

```
(%i11) g122:trigsimp(1/2*(-G[1,1])*diff(g[2,2],u));
```

```
(%o11) 
$$\frac{(b \cos(u) + a) \sin(u)}{b}$$

```

```
(%i12) g212:trigsimp(1/2*G[2,2]*diff(g[2,2],u));
```

```
(%o12) 
$$\frac{b \sin(u)}{b \cos(u) + a}$$

```

Визуализируем геодезические на торе  $a = 3, b = 2$ .

```
(%i1) torus:parametric_surface(
      (3+2*cos(u))*cos(v),
      (3+2*cos(u))*sin(v),
      2*sin(u), u, 0, 2*pi, v, 0, 2*pi)$
```

```

(%i2) u1:0$
curve1:parametric(
    (3+2*cos(u1))*cos(v1),
    (3+2*cos(u1))*sin(v1),
    2*sin(u1),v1,0,2*%pi)$

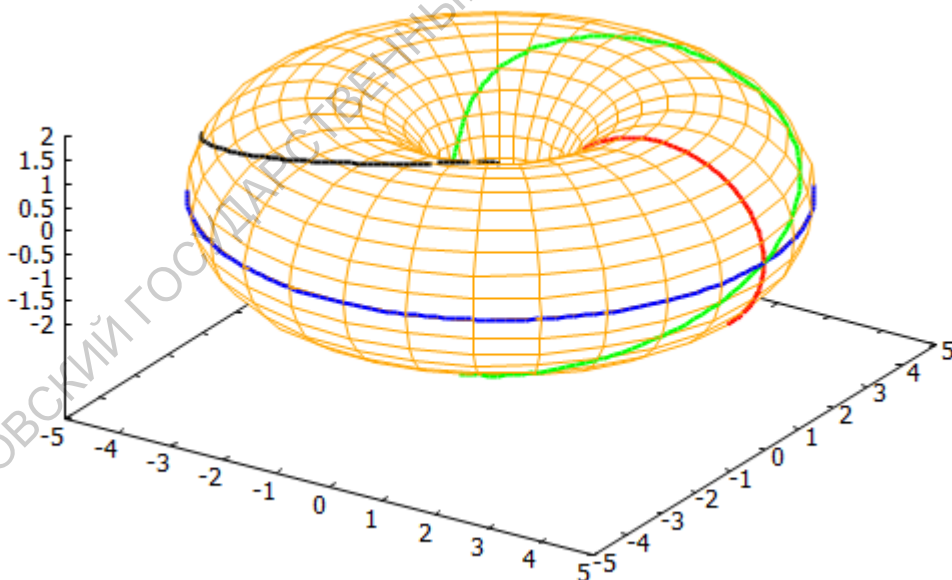
(%i4) v2:0$
curve2:parametric(
    (3+2*cos(u2))*cos(v2),
    (3+2*cos(u2))*sin(v2),
    2*sin(u2),u2,0,2*%pi)$

(%i6) t:2*%pi$
curve3:parametric(
    (3+2*cos(u3))*cos(u3+t),
    (3+2*cos(u3))*sin(u3+t),
    2*sin(u3),u3,0,2*%pi)$

(%i8) w:%pi$
curve4:parametric(
    (3+2*cos(v4+w))*cos(v4),
    (3+2*cos(v4+w))*sin(v4),
    2*sin(v4+w),v4,0,2*%pi)$

(%i10) load(draw)$
draw3d(nticks=200,surface_hide=true,
color=orange,torus,line_width=2,color=blue,curve1,
line_width=2,color=red,curve2,line_width=2,color=green,curve3,
line_width=2,color=black,curve4)$

```



19. Визуализировать геодезические на цилиндре:  $\vec{r} = \{a \cos \varphi, a \sin \varphi, b \varphi + c\}$ .

```
(%i1) a:1$
      b:1$
      c:1$
      p:11*%pi/6$

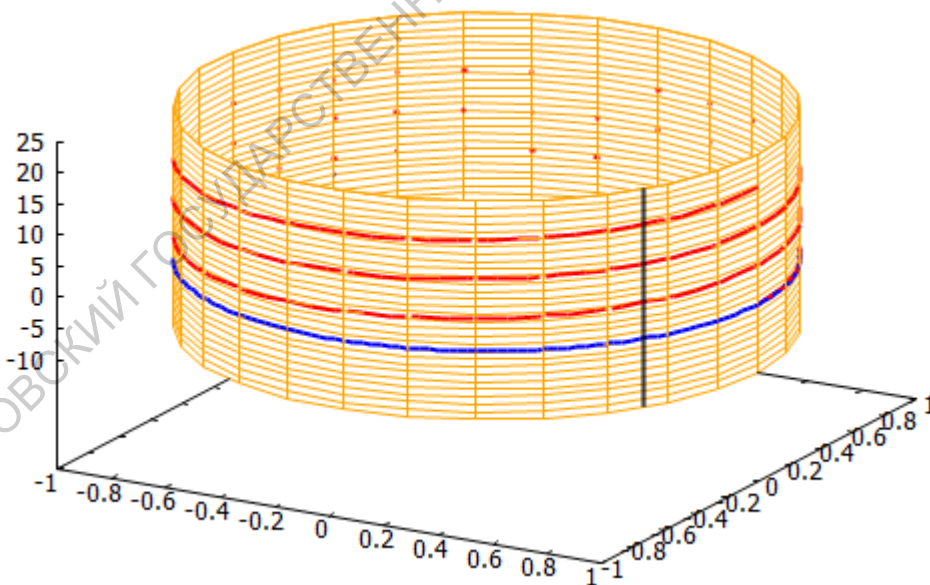
(%i5) cylindr:cylindrical(a,z,-10,25,phi,0,2*%pi)$

(%i6) curve1:parametric(
      a*cos(phi),
      a*sin(phi),
      a,phi,0,2*%pi)$

(%i7) curve2:parametric(
      a*cos(p),
      a*sin(p),
      z,z,-10,25)$

(%i8) curve3:parametric(
      a*cos(phi),
      a*sin(phi),
      b*phi+c,phi,0,6*%pi)$

(%i9) load(draw)$
      draw3d(nticks=200,surface_hide=true,color=orange,cylindr,
      line_width=2,color=blue,curve1,
      line_width=2,color=black,curve2,
      line_width=2,color=red,curve3)$
```

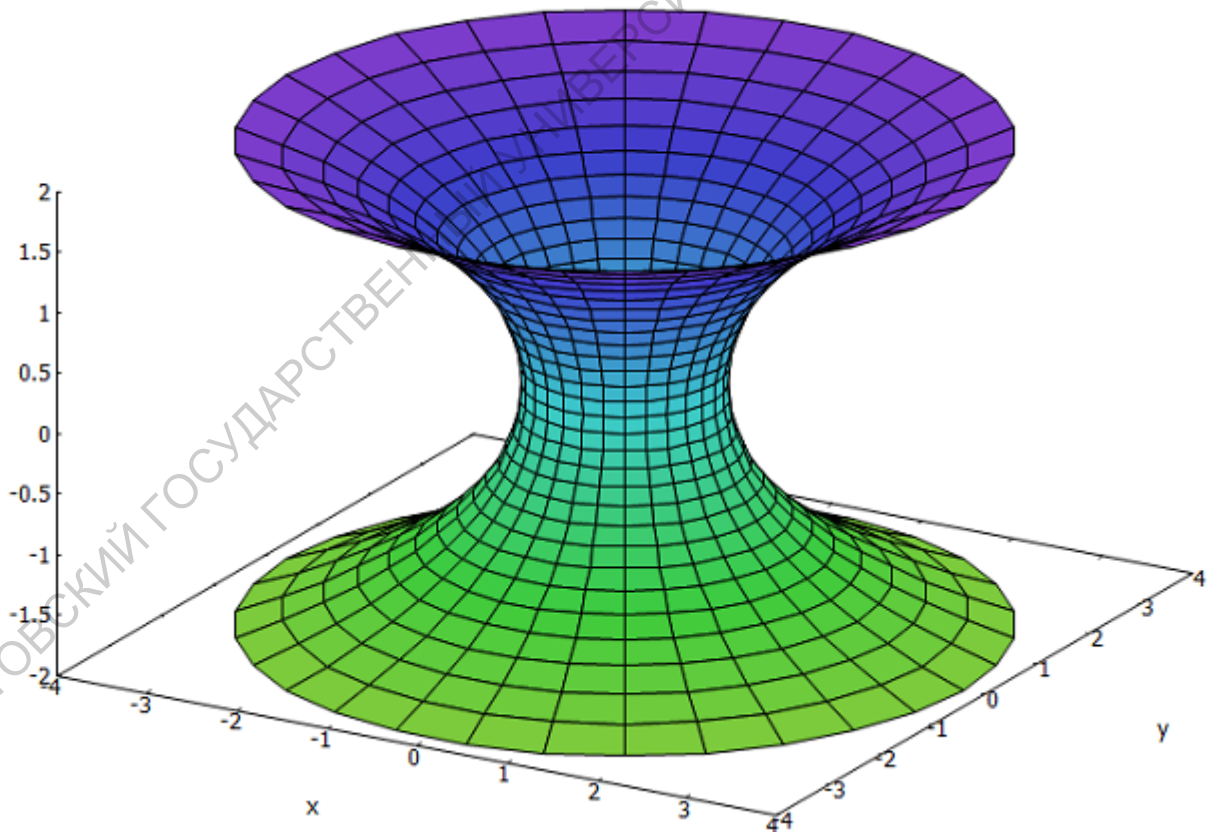


20. Изобразим изгибание катеноида в геликоид.

**Решение.** Изобразим катеноид (минимальная поверхность вращения):

```
(%i1) r(u,v):=[cosh(v)*cos(u), cosh(v)*sin(u), v];  
(%o1) r(u,v):=[cosh(v)cos(u), cosh(v)sin(u), v]  
  
(%i2) ru:diff(r(u,v),u);  
      rv:diff(r(u,v),v);  
(%o2) [-sin(u)cosh(v), cos(u)cosh(v), 0]  
(%o3) [cos(u)sinh(v), sin(u)sinh(v), 1]  
  
(%i4) E:trigsimp(ru.ru); F:ru.rv; G:trigsimp(rv.rv);  
(%o4) cosh(v)^2  
(%o5) 0  
(%o6) cosh(v)^2  
  
(%i7) plot3d(r(u,v), [u,0,2*%pi], [v,-2,2],  
            [gnuplot_pm3d,true])$
```

Parametric function



Зададим геликоид.

```

(%i8) r(u,v):=[sinh(v)*sin(u), sinh(v)*cos(u), u];
(%o8) r(u,v):=[sinh(v)sin(u),sinh(v)cos(u),u]

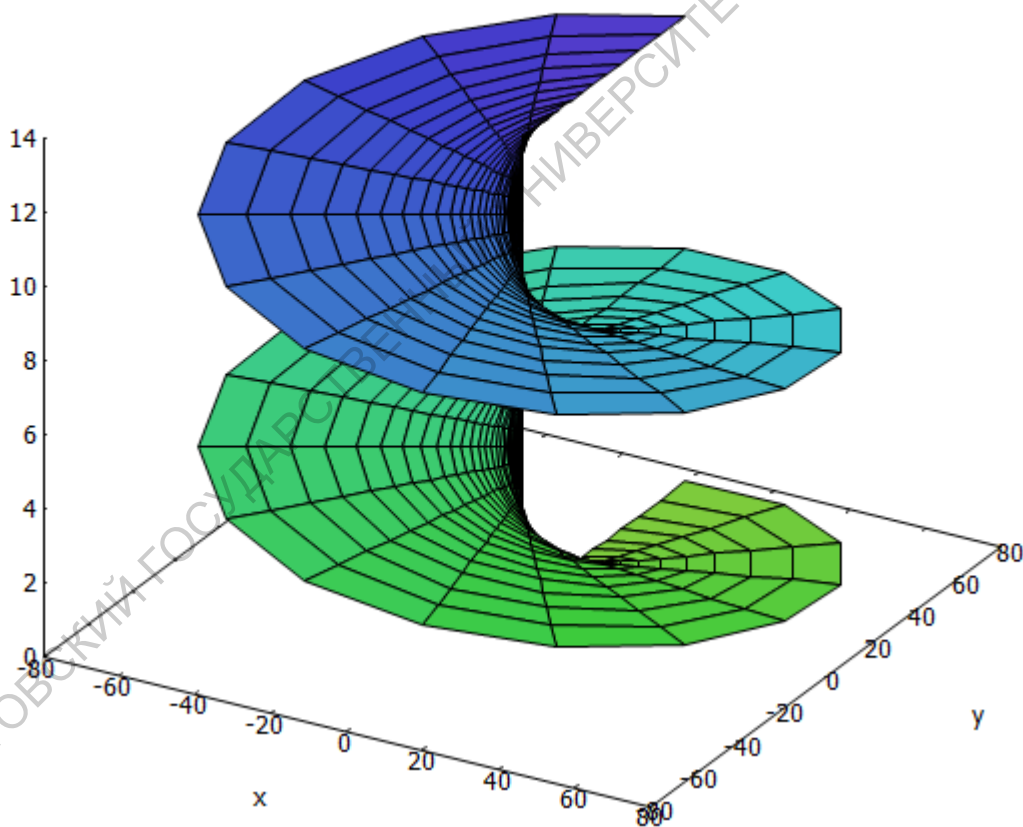
(%i9) ru:diff(r(u,v),u);
      rv:diff(r(u,v),v);
(%o9) [cos(u)sinh(v),-sin(u)sinh(v),1]
(%o10) [sin(u)cosh(v),cos(u)cosh(v),0]

(%i11) E:trigsimp(ru.ru); F:ru.rv; G:trigsimp(rv.rv);
(%o11) cosh(v)^2
(%o12) 0
(%o13) cosh(v)^2

(%i14) plot3d(r(u,v), [u,0,4*%pi], [v,0,5],
             [gnuplot_pm3d,true])$

```

Parametric function



```

(%i15) r(u,v,alpha):=[cos(alpha)*sinh(v)*sin(u)+sin(alpha)*
cosh(v)*cos(u),
-cos(alpha)*sinh(v)*cos(u)+sin(alpha)*cosh(v)*sin(u),
u*cos(alpha)+v*sin(alpha)];

(%o15) r(u,v,alpha):=[cos(alpha)sinh(v)sin(u)+sin(alpha)cosh(v)cos(u),
(-cos(alpha))sinh(v)cos(u)+sin(alpha)cosh(v)sin(u),u*cos(alpha)+v*sin(alpha)]

(%i16) ru:diff(r(u,v,alpha),u);
rv:diff(r(u,v,alpha),v);

(%o16) [cos(alpha)cos(u)sinh(v)-sin(alpha)sin(u)cosh(v),cos(alpha)sin(u)
sinh(v)+sin(alpha)cos(u)cosh(v),cos(alpha)]

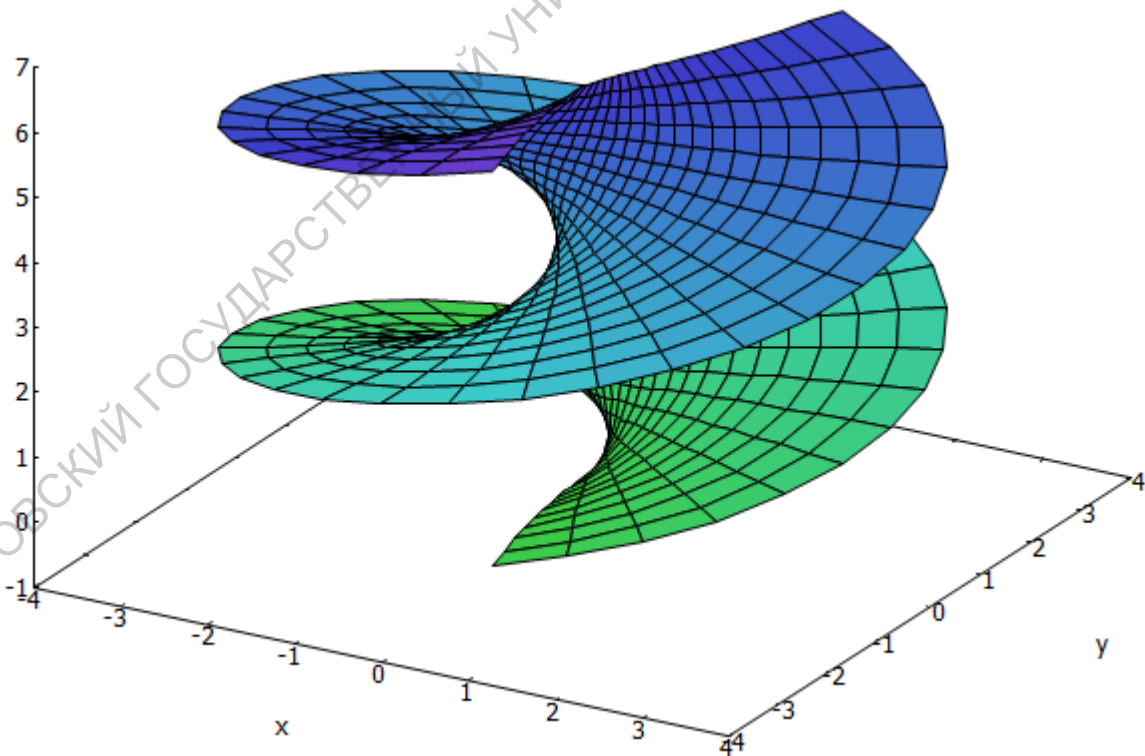
(%o17) [sin(alpha)cos(u)sinh(v)+cos(alpha)sin(u)cosh(v),sin(alpha)sin(u)
sinh(v)-cos(alpha)cos(u)cosh(v),sin(alpha)]

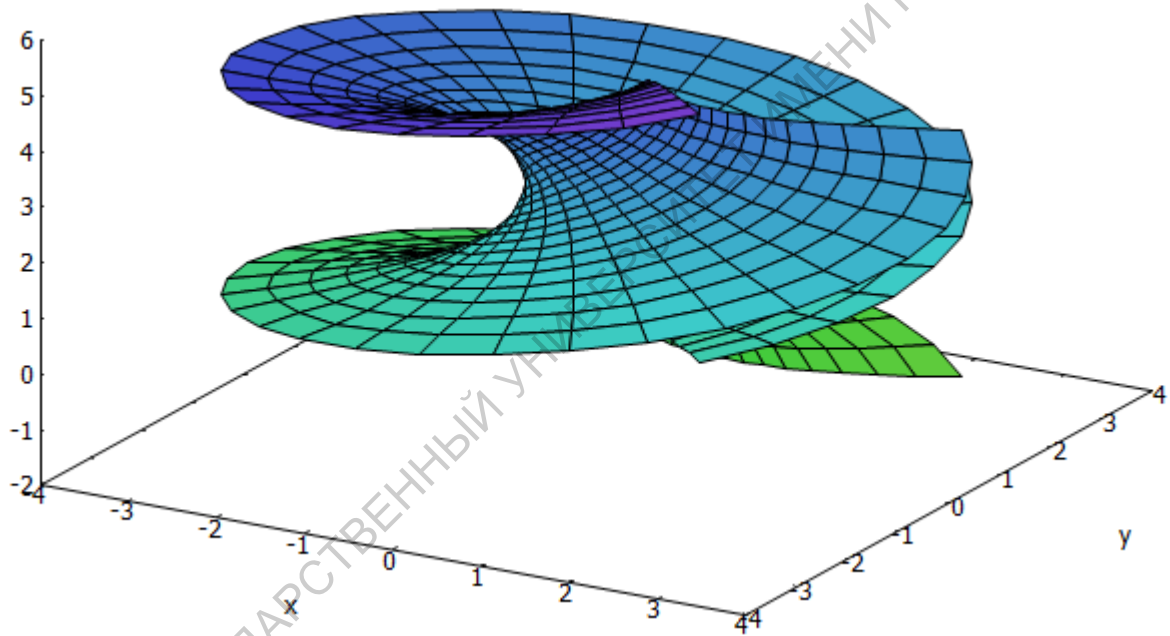
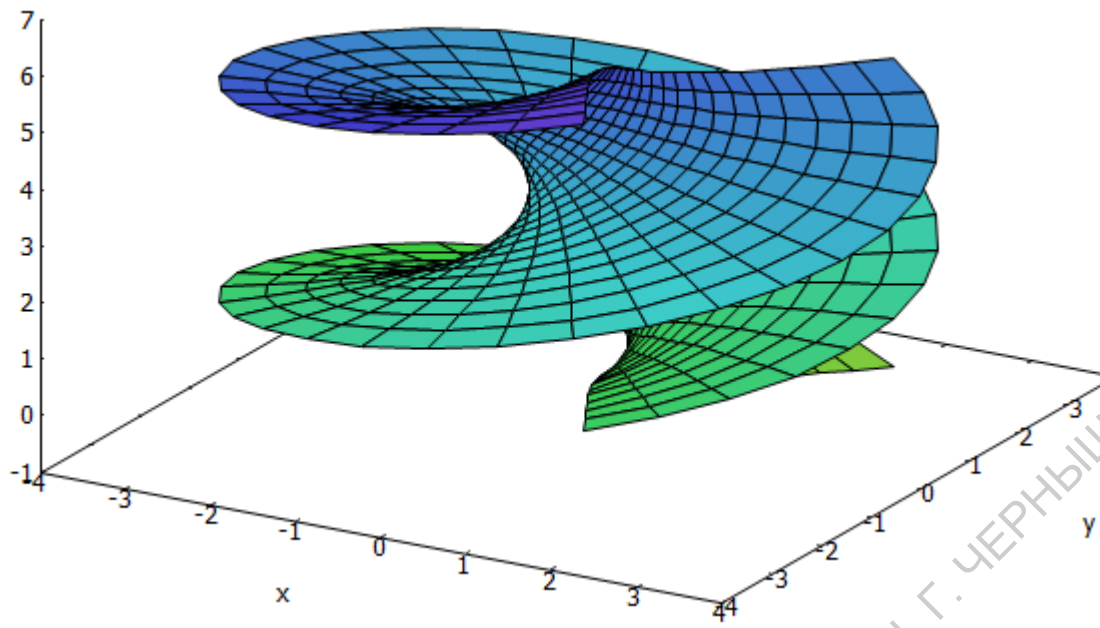
(%i18) E:trigsimp(ru.ru); F:trigsimp(ru.rv);
G:trigsimp(rv.rv);

(%o18) cosh(v)^2
(%o19) 0
(%o20) cosh(v)^2

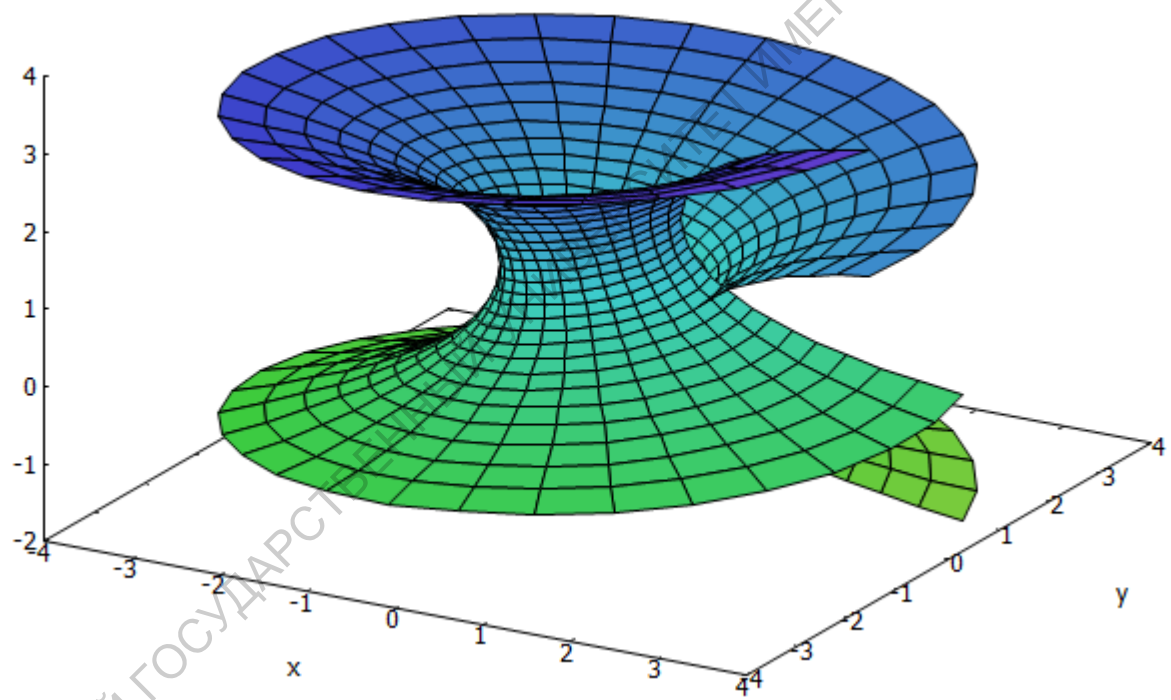
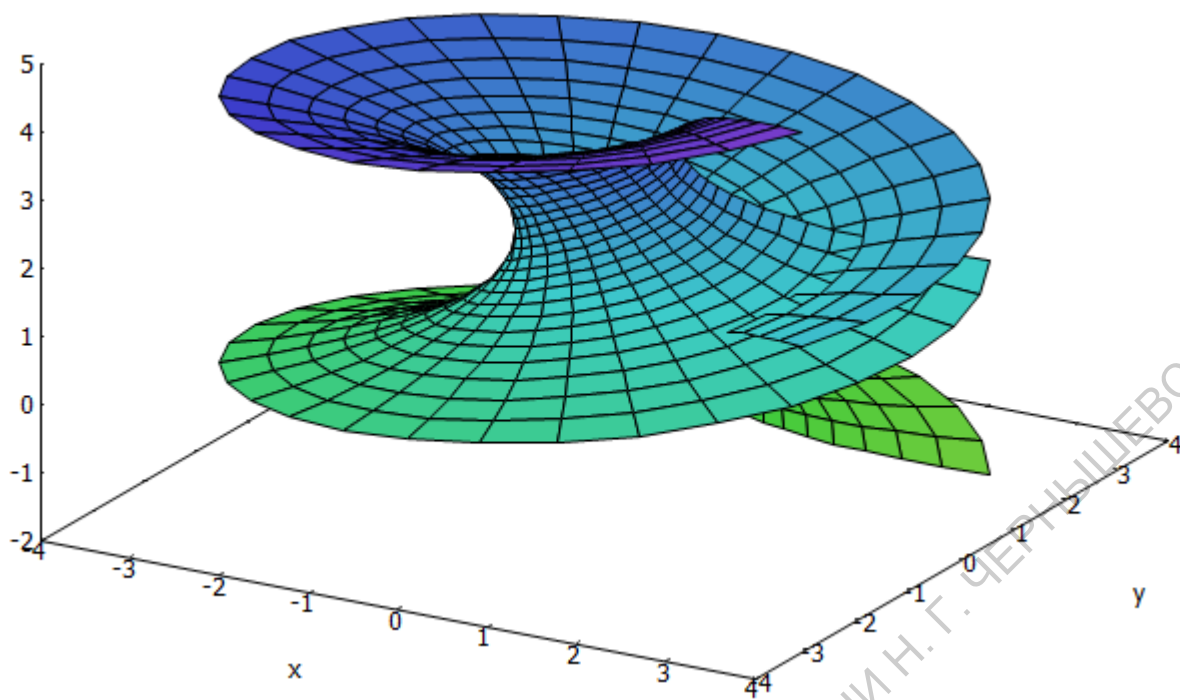
(%i21) for i from 0 thru 6 do(
alpha:%pi/12*i,
plot3d(r(u,v,alpha), [u,0,2*%pi], [v,-2,2]))$

```



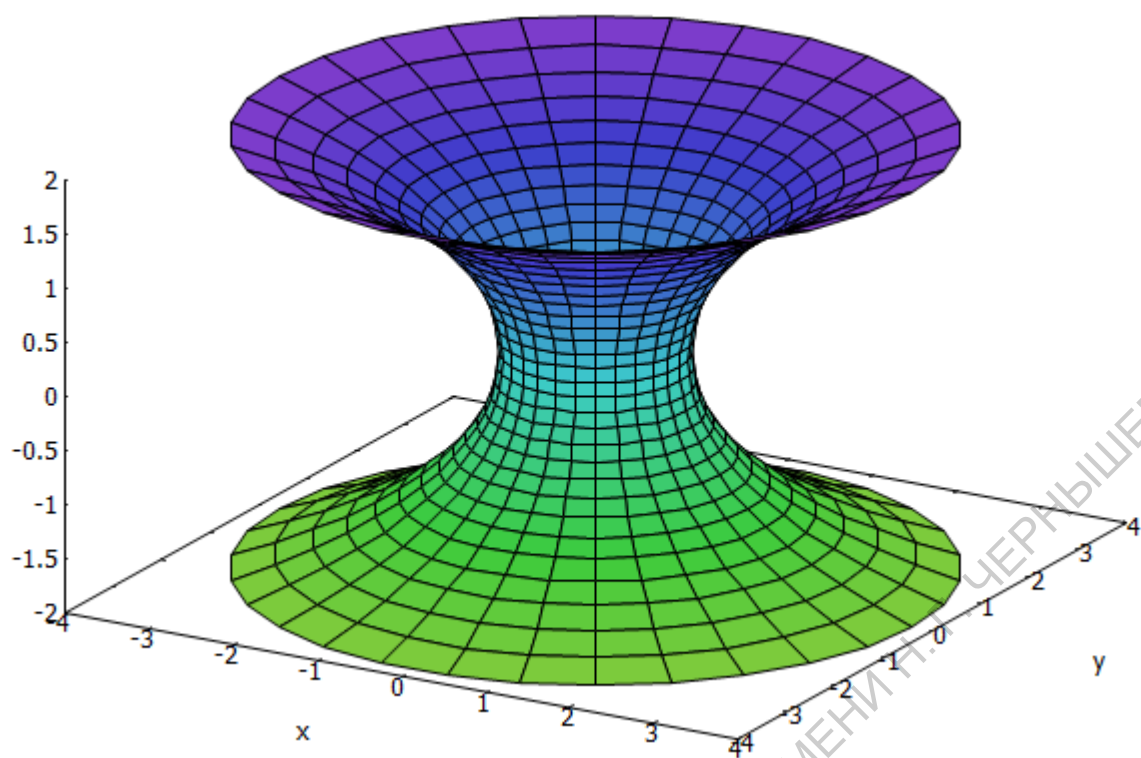


САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО



САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО





САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ П.А. ЧЕРНЫШЕВСКОГО

## Список литературы

1. Korol'kova A.V. Tensor computations in computer algebra systems [Электронный ресурс] / A.V. Korol'kova, D.S. Kulyabov, L.A. Sevast'yanov // arXiv:1402.6635 [cs.SC] [v1] Sat, 22 Feb 2014. 18 p. - Режим доступа: <http://arxiv.org/abs/1402.6635> (дата обращения: 01.10.2015)
2. Пипин В. Тензорная Алгебра в Maxima [Электронный ресурс] URL: <https://www.researchgate.net/publication/235217371TenzornaaAlgebravMaxima>
3. Подборка руководств по Maxima с официального сайта, в том числе и на русском языке. <http://maxima.sourceforge.net/ru/documentation.html>
4. Полищук Ю.В. Автоматизация символьных вычислений в документах LaTeX на основе Maxima [Электронный ресурс] / Ю.В. Полищук, Д.В. Пономарев // Открытый архив электронных изданий Оренбургского государственного университета. - Режим доступа: <http://elib.osu.ru/handle/123456789/327> (дата обращения: 01.10.2015)
5. Сборник задач по дифференциальной геометрии: учебное пособие для студентов / под ред. А. С. Феденко. - М.: Наука, 1979. - 272 с.
6. Чичкарёв Е. А. Компьютерная математика с Maxima [Электронный ресурс] / Е.А. Чичкарёв // ИНТУИТ - Режим доступа: <http://www.intuit.ru/studies/courses/3484/726/info> (дата обращения: 01.10.2015)

## Приложение

### Преобразование арифметических выражений

`factor` - разложение на множители

`ratsimp` - упрощение выражения. Функция `fullratsimp` вызывает функцию `ratsimp` до тех пор, пока выражение не перестанет упрощаться.

Если требуется вычислить числовое значение, полученного выражения, то можно применить функцию `at`, указав в скобках выражение или его адрес и значение переменной.

`divide` - нахождение частного и остатка от деления одного многочлена на другой

Первый элемент полученного массива - частное, второй - остаток от деления. `gcd` - наибольший общий делитель многочленов

`partfrac` - преобразует в простые дроби по заданной переменной

`radcan` - преобразует выражения, содержащие логарифмические, показательные и степенные функции.

Для упрощения тригонометрических и гиперболических выражений применяется приставка `trig`. Для наилучшего результата ее можно комбинировать с функциями `ratsimp`, `fullratsimp`, `radcan` и другими. Команда `trigsimp()` пытается упростить выражение, применяя к нему простейшие тригонометрические тождества

`trigexpand` - тригонометрическое раскрытие скобок. Использует формулы преобразования сумм двух углов для представления введенного выражения в как можно более простом виде - где в качестве аргумента только одна переменная

Команда `trigreduce()` выполняет свертывание всех произведений тригонометрических функций в тригонометрические функции от сумм.

## Решение уравнений и систем алгебраических уравнений

Решение алгебраических уравнений и их систем осуществляется при помощи функции `solve`, в качестве параметров. В первых квадратных скобках указывается список уравнений через запятую, во вторых - список переменных, через запятую (либо несколько упрощённые формы записи):

`solve(expr, x)` - решение одного уравнения относительно переменной  $x$ ;

`solve([eqn1, . . . , eqnn], [x1, . . . , xn])` - решение системы уравнений.

`lhs(eqn)` - левая часть уравнения `eqn`.

`rhs(eqn)` - правая часть уравнения `eqn`.

`allroots(polyn)` - численные приближения многочлена от одной переменной.

Функции для работы с матрицами

Оператор	Описание
$A+B$	Сложение матриц A и B
$A-B$	Разность матриц A и B
$s*A$	Умножение матрицы A на число s
$A.B$	Умножение матриц A и B
$A^n$	n-ая степень матрицы A
$A^{-1}$	обратная матрица для матрицы A
<u>invert(A)</u>	обратная матрица для матрицы A
<u>determinant(A)</u>	определитель матрицы A
<u>eigenvalues(A)</u>	нахождение собственных значений матрицы
<u>minor(A, i, j)</u>	минор матрицы A, i – номер строки, j – номер столбца
<u>rank(A)</u>	ранг матрицы A
<u>transpose(A)</u>	транспонирование матрицы A
<u>submatrix(i, A, j)</u>	возвращает матрицу, полученную из исходной удалением i строки и (или) j столбца.
<u>diagmatrix(n, x)</u>	Диагональная матрица порядка n с элементами x
<u>ident(n)</u>	Единичная матрица n-го порядка

Построение трёхмерных графиков

Основная команда для построения трёхмерных графиков - `plot3d`.

Цвет поверхности задается опцией `palette`: `[palette, [palette1], ... [paletten]]`.

Команда `palette` состоит из списка, включающего ключевое слово, за которым следуют четыре числа. Ключевое слово - это `hue` (оттенок), `saturation` (насыщенность) или `value` (значение). Первые три цифры, которые должны быть между 0 и 1, определяют оттенок, насыщенность и значение основных цветов.

Ключевое слово определяет, какой из трех атрибутов (оттенок, насыщенность или стоимости) будет увеличена в соответствии со значениями `g`. Последняя цифра указывает на рост, соответствующий максимальному значению.

Это последнее число может быть больше, чем 1 или отрицательное; соответствующие значения измененных атрибутов будут округлены по модулю 1.

Цвет поверхности задаются опции палитры. Каждая палитра список ключевых слов следуют четыре цифры. Ключевое слово определяет, какой из трех атрибутов (оттенок, насыщенность или стоимости) будет увеличена в соответствии со значениями. Первые три цифры, которые должны быть между 0 и 1, определяют основу цвет, который будет назначен до минимального значения `g`. Последняя цифра указывает на рост, соответствующий максимальному значению `g`.

По умолчанию выполняется: `[hue, 0.25, 0.7, 0.8, 0.5]`

Построение графических иллюстраций: пакет draw

В Maxima имеется несколько альтернативных библиотек для отображения графиков функций, наборов точек, трехмерных тел, градиентов и т.д. По умолчанию используется библиотека plot, но для решения некоторых задач может оказаться удобнее библиотека draw. Пакет draw обеспечивает более мощное и гибкое построение рисунков. Он поддерживает только gnuplot. Перед использованием draw необходимо загрузить командой load("draw").

Графические объекты в 2d

explicit(func, x, xmin, xmax) - явная функция от переменной x;

xrange=[xmin, xmax] - диапазон горизонтальной оси (по умолчанию - автоматически);

yrange=[ymin, ymax] - диапазон вертикальной оси (по умолчанию - автоматически);

color - цвет кривой (по умолчанию - цвет кривой синий);

implicit(feq, x, xmin, xmax, y, ymin, ymax) - неявная функция, заданная уравнением feq с переменными x и y в 2D;

parametric(xfun, yfun, t, tmin, tmax) - параметрические функции с параметром t в 2D;

nticks=integer - количество точек, используемых для построения (по умолчанию - 29)

points([xcoords,...],[ycoords,...]) - точки

Графические объекты в 3d

explicit(func, x,xmin,xmax, y,ymin,ymax) - явная функция с переменными x, y;

surfacehide - если true, то невидимая часть поверхности не рисуется (по умолчанию false);

enhanced3d - если true, то цветная поверхность;

colorbox - цветовая шкала;

implicit(feq, x,xmin,xmax, y,ymin,ymax, z,zmin,zmax) - неявная функция;

parametric(xfun,yfun,zfun, t,tmin,tmax) - параметрические функции;

parametricsurface(xfun,yfun,zfun, s,smin,smax, t,tmin,tmax) - параметрические функции.

[color, color1, color2,...] - цвет линии, точки

## Вычисление производных функций

Производная функции  $f(x)$  вычисляется с помощью оператора  $\text{diff}(f(x), x, n)$ ; Здесь  $n$  - это порядок производной.

Для вычисления первой производной вводится команда:  $\text{diff}(f(x), x, 1)$  или  $\text{diff}(f(x), x)$

Если указать апостроф перед символом  $\text{diff}$ , то производная не вычисляется.

С помощью функции  $\text{diff}$  можно вычислять частные производные функции многих переменных:

$\text{diff}(f(x,y), x, n, y, m)$ ;

где  $n$  - порядок производной по переменной  $x$ ,  $m$  - порядок производной по переменной  $y$ .

## Аналитическое и численное интегрирование

Неопределённый интеграл вычисляется с помощью команды

$\text{integrate}(f(x), x)$ ,

где  $f(x)$  - подынтегральная функция,  $x$  - переменная интегрирования.

Для вычисления определённого интеграла в команде  $\text{integrate}$  добавляются пределы интегрирования  $\text{integrate}(f(x), x, \text{нижний предел}, \text{верхний предел})$ .

Если результат зависит от значений постоянных, Maxima спросит об этом пользователя. В следующем примере, необходимо ввести "p"(positive) в случае  $a > 0$  или "n"(negative) в случае  $a < 0$ .

Некоторые интегралы Maxima может записать через специальные функции. Для вычисления численного значения таких интегралов используется команда  $\text{numer}$ .

Если определённый интеграл не вычисляется, то Maxima просто запишет его в символьном виде. Можно найти приближенное значение интеграла численными методами. Это можно сделать командой  $\text{quad}_{ags}()$ .



Решение дифференциального уравнения

Функция `ode2` позволяет решить обыкновенные дифференциальные уравнения (ОДУ) первого и второго порядков.

`ode2(eq, dvar, ivar)` - решение ОДУ первого и второго порядков. `eq` - выражение, определяющее дифференциальное уравнение, `dvar` - зависимая переменная, `ivar` - независимая переменная

`ic1(sol, xval, yval)` - решение задачи Коши первого порядка

`ic2(sol, xval, yval, dval)` - решение задачи Коши второго порядка

Посредством функции `ode2` могут быть решены следующие типы ОДУ первого порядка: линейные, ОДУ с разделяющимися переменными, однородные ОДУ, уравнения в полных дифференциалах, уравнения Бернулли, обобщённые однородные уравнения.

Дифференциальное уравнение представляется в форме производной, вычисление которой запрещено с помощью одиночной кавычки: `'diff(y, x)`. Другой вариант явно указать зависимость  $y = y(x)$  - использовать функцию `depends` (в этом случае можно не использовать начальный апостроф). Если `ode2` не может получить решение, она возвращает значение `false`.

`desolve(eq, y)` - решение линейного ОДУ

`desolve([eq1, ..., eqn], [y1, ..., yn])` -

`atvalue(expr, [x1 = a1, ...], c)` -  $cx_1 = a_1$ .

При использовании `desolve` необходимо явно задавать функциональные зависимости (вместо `'diff(y, x)` использовать запись `diff(y(x), x)`).

Для численного решения дифференциальных уравнений и систем необходимо предварительно загрузить пакет `dynamics`: `load(dynamics)`.

Численное интегрирование дифференциального уравнения методом Рунге - Кутты выполняется командой `rk()`. Синтаксис вызова:

`rk([eq], [vars], [init], [trange])`,

где `eq` - список правых частей уравнений; `vars` - список зависимых переменных; `init` - список начальных значений; `trange` - список `[t, t0, tend, ht]`, содержащий символьное обозначение независимой переменной (`t`), её начальное значение (`t0`), конечное значение (`tend`), шаг интегрирования (`ht`).

Системы дифференциальных уравнений решаются с использованием той же команды `rk()`. Уравнения, искомые переменные и начальные условия к ним перечисляются в квадратных скобках. Уравнения, искомые переменные и начальные условия к ним перечисляются в квадратных скобках.

Программирование на встроенном макроязыке

Условный оператор в Maxima имеет вид

```
if условие then действие1 else действие2;
```

Если условие истинно, то выполняется выражение действие1, иначе - выполняется выражение действие2.

```
if условие then действие;
```

Если условие истинно, то выполняется выражение действие.

```
if условие1 then действие1 else
```

```
if условие2 then действие2 else
```

```
if...else действиеN
```

Если выполняется условие 1, то выполняется выражение действие1, иначе - проверяется условие2, и если оно истинно - выполняется выражение действие 2, и т.д. Если ни одно из условий не является истинным - выполняется выражение действие N.

Виды циклов:

```
for i: i1 step i2 thru i3 do тело цикла;
```

```
for i: i1 step i2 while условие do тело цикла
```

```
for i: i1 step i2 unless условие do тело цикла
```

Здесь  $i$  - переменная цикла;  $i1$  - начальное значение;  $i2$  - шаг (по умолчанию равен 1);  $i3$  - конечное значение переменной цикла; тело цикла - операторы тела цикла.  $i1$ ,  $i2$ ,  $i3$  и тело цикла могут быть произвольными выражениями.

Ключевые слова `thru`, `while`, `unless` указывают на способ завершения цикла: по достижении переменной цикла значения  $i3$ ; пока выполняется условие `условие`; пока не будет достигнуто условие `условие`.

При нормальном завершении цикла на экран возвращается `done`. Принудительный выход из цикла осуществляется при помощи оператора `return`, который может возвращать произвольное значение.

Массивы - это упорядоченная последовательность величин (элементов массива), обозначаемая одним именем. Прежде чем создать массив его нужно описать с помощью функции `array`

```
array (name, dim1, ..., dimn)
```

```
array (name, type, dim1, ..., dimn)
```

```
array ([name1, ..., namem], dim1, ..., dimn)
```

Здесь `name` - это имя массива,  $dim_1, \dots, dim_n = .. - , 0dim_i, \dots, .$