

СОЗДАНИЕ БИБЛИОТЕКИ ПРОЦЕДУР В ДИФФЕРЕНЦИАЛЬНОЙ ГЕОМЕТРИИ

Игнатъев Ю.Г., Самигуллина А.Р.

E-mail: ignatev_yu@rambler.ru, Alsu_Sam@rambler.ru

Татарский государственный гуманитарно-педагогический университет, г. Казань

Аннотация. Рассмотрены проблемы создания пользовательской библиотеки дифференциальной геометрии в пакете Maple.

Construction of procedures library on differential geometry

Ignatyev Yu., Samigullina A.

Abstract. The problems of construction user's library on Differential Geometry by tools package Maple are considered.

Дифференциальная геометрия традиционно считается одним из наиболее трудных предметов на математических факультетах вузов. В ней интегрируются знания, полученные студентами на курсах алгебры, аналитической геометрии, математического анализа одной и нескольких переменных, дифференциальных уравнений. Кроме того, геометрические образы, а также соответствующие вычислительные процедуры, сопровождающие этот курс, являются чрезвычайно сложными и громоздкими. Все это делает указанный курс мало наглядным и сложным для усвоения, несмотря на чрезвычайную простоту и прозрачность основных идей дифференциальной геометрии. С другой стороны, методы дифференциальной геометрии кривых и поверхностей находят огромное количество приложений в самых различных областях прикладной математики, теории поля, механики, оптики и т.п. Поэтому, задача повышения наглядности этого курса становится чрезвычайно важной. Эта задача может быть решена графическими средствами символьной математики, в частности, пакета Maple. При попытке создания достаточно качественных демонстраций по темам: соприкосновение кривых и поверхностей, натуральный репер кривой, натуральный репер на поверхности, нормальная и геодезическая кривизна поверхностей, геодезические линии на поверхностях и др. мы сталкиваемся с необходимостью создания специальных вычислительных и графических процедур, соответствующих вводимым понятиям, и создания анимационных процедур для демонстрации основных положений и результатов дифференциальной геометрии.

В докладе описана пользовательская библиотека процедур вычисления и графического отображения основных понятий и процедур дифференциальной геометрии кривых и поверхностей, приведены иллюстрации применения процедур на примерах различных кривых и поверхностей. Продемонстрируем некоторые процедуры этой библиотеки.

Будем задавать параметризованную кривую правой частью ее параметрических уравнений, т.е., радиусом-вектором $[f(t), g(t), h(t)]$, Line - будет ее условное имя, param - ее параметр.

Зададим для удобства в дальнейшем команду вычисления длины вектора, ModVector(x) и команду нормирования вектора, NormVector(x):

```
> restart:
> YuDifGeo[ModVector]:=proc(x):
simplify(sqrt(linalg[innerprod](x,x))):end:
YuDifGeo[NormVector]:=proc(x):
simplify(linalg[scalarmul](x,1/YuDifGeo[ModVector](x))):end:
```

Определим команду вычисления касательного вектора к кривой в точке с параметром param0:

```
> YuDifGeo[TangentLine]:=proc(Line,param,param0):
eval(subs(param=param0,diff(Line,param))):end:
> YuDifGeo[TangentLine]([a*cos(t),a*sin(t),b*t],t,0);
```

Определим команду вычисления длины дуги кривой:

```
> YuDifGeo[LengthLine]:=proc(Line,param,param0,param1)
local t:
simplify(int(YuDifGeo[ModVector](YuDifGeo[TangentLine](Line,param,t1)),
t1=param0..param1)):end:
> YuDifGeo[LengthLine]([a*cos(t),a*sin(t),b*t],t,0,t);
```

Определим команду вычисления единичного касательного вектора кривой tau в точке с параметром param0:

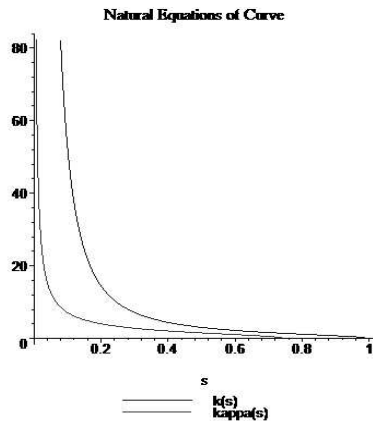
```
> YuDifGeo[NormTangentLine]:=proc(Line,param,param0):
YuDifGeo[NormVector](YuDifGeo[TangentLine](Line,param,param0)):end:
> YuDifGeo[NormTangentLine]([a*cos(t),a*sin(t),b*t],t,t);
```

Определим команду нахождения единичного вектора бинормали: beta= $[r',r''] / |[r',r'']|$:

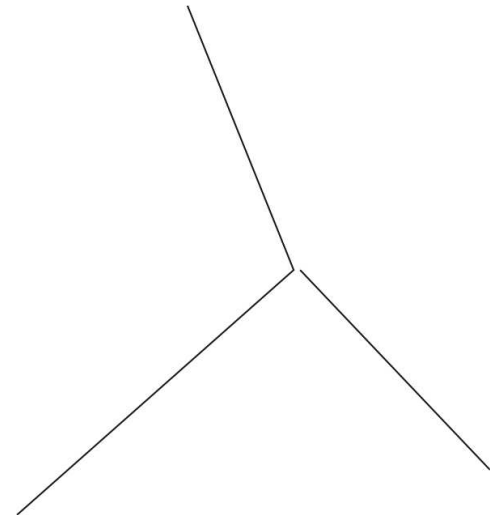
```

> YuDifGeo[BinormLine]:=proc(Line,param,param0):
  YuDifGeo[NormVector](linalg[crossprod](eval
(subs(param=param0,diff(Line,param))),
eval(subs(param=param0,diff(Line,param$2))))):end:
> YuDifGeo[BinormLine]([a*cos(t),a*sin(t),b*t],t,t);
Определим команду нахождения единичного вектора главной нормали  $n=[\tau,\beta]$ :
> YuDifGeo[MainNormLine]:=proc(Line,param,param0):
  simplify(linalg[crossprod](YuDifGeo[NormTangentLine](Line,param,param0),
YuDifGeo[BinormLine](Line,param,param0))):end:
> YuDifGeo[MainNormLine]([a*cos(t),a*sin(t),b*t],t,t);
Определим команду вычисления кривизны кривой в точке param0:
> YuDifGeo[CurvatureLine]:=proc(Line,param,param0) local T1,N1:
  T1:=subs(param=param0,diff(Line,param)):N1:=subs
(param=param0,diff(Line,param$2)):
  simplify(YuDifGeo[ModVector](linalg[crossprod]
(T1,N1))/YuDifGeo[ModVector](T1^3)):
  end:
> YuDifGeo[CurvatureLine]([a*cos(t),a*sin(t),b*t],t,tau);
Определим команду смешанного произведения трех векторов:
> YuDifGeo[CombProd]:=proc(x,y,z):
  simplify(linalg[innerprod](x,linalg[crossprod](y,z))):end:
> YuDifGeo[CombProd]([x,y,z],[1,2,3],[4,5,6]);
Определим команду вычисления кручения кривой в точке param0:
> YuDifGeo[TorsionLine]:=proc(Line,param,param0) local T1,T2,T3,TTT:
  T1:=subs(param=param0,diff(Line,param)):T2:=subs
(param=param0,diff(Line,param$2)):
  T3:=subs(param=param0,diff(Line,param$3)):TTT:=
  simplify(YuDifGeo[ModVector](linalg[crossprod](T1,T2))):
  simplify(YuDifGeo[CombProd](T1,T2,T3)/TTT^2):end:
Команды изображения в теории дифференциальной геометрии кривых.
Команда изображения натуральных уравнений кривых  $k=k(s)$ ;  $\kappa=\kappa(s)$ :
> YuDifGeo[GraphicNaturalCurvature]:=proc
(Line,param,param0,max_param,cc)
  local param1:
  plot([YuDifGeo[CurvatureLine](Line,param,param1),
YuDifGeo[LengthLine](Line,param,param0,param1),
param1=param0..max_param],color=cc
,labelfont=[TIMES,ROMAN,12],labels=['s','']):end:
  YuDifGeo[GraphicNaturalTorsion]:=proc (Line,param,param0,max_param,cc)
  local param1:
  plot([YuDifGeo[TorsionLine](Line,param,param1),
YuDifGeo[LengthLine](Line,param,param0,param1),
param1=param0..max_param], color=cc,
labelfont=[TIMES,ROMAN,12],labels=['s','']):end:
  > YuDifGeo[GraphicNaturalEquations]:= proc (Line,param,param0,max_param,cc1,cc2)
  local param1: plot([YuDifGeo[CurvatureLine](Line,param,param1),
YuDifGeo[LengthLine](Line,param,param0,param1), param1=param0..max_param],
[YuDifGeo[TorsionLine](Line,param,param1),
YuDifGeo[LengthLine](Line,param,param0,param1), param1=param0..max_param]), color =
[cc1,cc2], labelfont = [TIMES,ROMAN,12], labels=['s',''], legend=['k(s) ','kappa(s)'],
titlefont = [TIMES,ROMAN,BOLD],
title='Natural Equations of Curve':end:
  > YuDifGeo[GraphicNaturalEquations]([t*cos(t),t*sin(t),t],t,0,4*Pi,blue,red);
(Рис. 1.а.)
Команда изображения отрезка
> YuDifGeo[ParamLine]:=proc(X,q,param,c) local EQ:
EQ:=linalg[matadd](X,linalg[scalarmul](q,param),1,1):
plots[spacecurve](EQ,param=0..1,thickness=2,color=c,scaling=CONSTRAINED):
end:
Команды изображения адаптированного репера с началом в точке t0:
> YuDifGeo[NaturalReper]:=proc(Line,param,param0,cc1,cc2,cc3)
local rt1,rt2,rt3,tt,bb,nn,xi,zeta,chi:

```



а. Графическое представление натуральных уравнений кривой. Вдоль оси абсцисс отложены значения натурального параметра s .



б. Визуализация натурального репера в заданной точке кривой.

Рис.1. Графические представления результатов исполнения процедур.

```

tt:=YuDifGeo[NormTangentLine](Line,param,param0):
rt1:=YuDifGeo[ParamLine](subs(param=param0,Line),
tt,xi,cc1,scaling=CONSTRAINED):
bb:=YuDifGeo[BinormLine](Line,param,param0):
rt2:=YuDifGeo[ParamLine](subs(param=param0,Line),
bb,zeta,cc2,scaling=CONSTRAINED):
nn:=YuDifGeo[MainNormLine](Line,param,param0):
rt3:=YuDifGeo[ParamLine](subs(param=param0,Line),nn,
chi,cc3,scaling=CONSTRAINED):
plots[display](rt1,rt2,rt3):end:
> YuDifGeo[NaturalReper]([cos(t),sin(t),t],t,Pi/2,red,blue,black);
Команда анимации натурального репера кривой:
> YuDifGeo[AnimNaturalReper]:=proc(Line,param,mini,maxi) local GL,S:
GL:=plots[spacecurve](Line,param=mini..maxi,
thickness=1,color=navy,scaling=CONSTRAINED,
numpoints=2000):
S:=(i)->plots[display](YuDifGeo[NaturalReper](Line,param,
(maxi-mini)/71*i,black,blue,red),GL):plots[display](seq(S(i),i=0..71),insequence=true):
end:
> YuDifGeo[AnimNaturalReper]
([cos(t),sin(t),0.2*t],t,0,8*Pi);

```

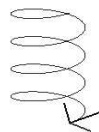


Рис.2. Визуализация анимационной процедуры движения натурального репера вдоль винтовой линии.

Восстановление кривой по натуральным уравнениям (Метод Игнатъева).

Зададим единичный вектор касательной к кривой в сферических координатах:

```

> YuDifGeo[TangentSpher]:=proc(Phi,Theta,natural_param):
[cos(Phi(natural_param))*cos(Theta(natural_param)),
 sin(Phi(natural_param))*cos(Theta(natural_param)),sin
(Theta(natural_param))]:end:
> YuDifGeo[TangentSpher](phi,theta,s);
> YuDifGeo[kSpher]:=proc(Phi,Theta,natural_param):
simplify(YuDifGeo[ModVector](diff(YuDifGeo[TangentSpher]
(Phi,Theta,natural_param),
natural_param)))]:end:
> YuDifGeo[kSpher](phi,theta,s);

```

Литература

1. Дьяконов В. «Maple7:учебный курс». — СПб.: Питер,2002.;
2. Игнатъев Ю.Г. «Дифференциальная геометрия кривых и поверхностей в евклидовом пространстве ». - Учебное пособие. Казань: ТГГПУ, 2006;
3. Игнатъев Ю.Г., Самигуллина А.Р. Методики и технологии математического образования. Сборник трудов Международной научной конференции «Математика. Образование. Культура» - г. Тольятти, 2005.