# Методический материал

## Научно-практические занятия по геометрии для студентов 3-го года кафедры Высшей геометрии и топологии

### Введение

Нет смысла говорить, насколько плотно компьютеры входят в повседневную жизнь математика, и еще более очевидно, что они не в состоянии заменить современного научного деятеля, который должен решать все более и более абстрактные задачи. В этой связи основная задача практикума:

*дать представление о принципиальных возможностях автоматизированных пакетов типа Wolfram Mathematica, то есть продемонстрировать с одной стороны универсальность их применения для широкого класса задач, но с другой стороны показать, что роль исследователя является незаменимой при интерпретации результатов, полученных автоматизировано.*

Практикум из 9 занятий построен так, что в его ходе решаются частные модельные задачи, позволяющие закрепить знания по пакету основных средств среды, а также разрабатываются индивидуальные проекты, связанные с научными задачами, поставленными руководителями участников практикума. Несмотря на то, что в основном работа происходит с пакетом Wolfram Mathematica, участники не ограничены его средствами. Некоторые студенты использовали другие специализированные пакеты (напр. Macauley) и общие языки программирования высокого уровня (C++).

Существенный акцент в построении занятий ставился на научной составляющей работы: прежде всего на навыках постановки научной задачи, умении представить научный проект, и, в последствии, на способности интерпретировать результат работы.

Ниже приводится перечень основных закрепляемых навыков, указывается в какой задаче и каким образом это производится. Далее методический материал содержит описание задач, модельных в разделе 1 и индивидуальных в разделе 2, с детальным описанием решения и комментариями об используемых методах.

**Основные закрепляемые навыки:**

|  |  |  |
| --- | --- | --- |
| **Тема** | **Функции** | **Задача** |
| Работа со списками | Join, Flatten, Partition,  | 1.2, 1.3, 2.1, 2.2, 2.3 |
| Решение уравнений | Операции с массивом решений | 1.1, 2.2 |
| Решение дифференциальных уравнений | Численное решение уравнений | 1.4,  |
| Упрощение аналитических выражений | Expand, Together, Factor, | 2.1 |
| Программирование | Булевы операции, условные операторы, циклы, локализация переменных | 1.1, 1.2, 1.3, 2.1, 2.2, 2.3 |
| Элементы анимации | Slider, Dynamic, Slider2D, Manipulate, Locator | 1.1, 1.2, 1.4, 2.3 |
| Элементы графики | Plot, Plot3D, ParametricPlot, ParametricPlot3D | 1.1, 1.2, 1.4, 2.3 |
|  |  |  |

### Модельные задачи

#### Закон сложения на эллиптической кривой

Демонстрация закона ассоциативности групповой операции на торе.

Рассмотрим тор ${C}/{Z^{2} .}$ Зададим его сечение уравнением $ y^{2}=x^{3}+b\_{1}x+b\_{2}. $Операцию зададим так: выберем точку, которую будем считать единицей $Е=(xo,yo)$. Она изображена красным цветом. Для того, чтобы перемножить две точки А и В, проведем через них прямую, получим точку С (розовые точки на рисунке), являющуюся пересечением прямой АВ с кубической кривой, отличную от А и В. Затем проведем прямую через точки Е и С. Точка, полученная как пересечение прямой ЕС и отличная от Е и С(на рисунке салатовые), и будет произведением точек А и В. Итак, точки, которые мы перемножаем, имеют координаты (x1,y1), (x2,y2), (x3,y3). Они изображены черным цветом. Обозначим операцию $×$. Тогда мы хотим получить следующее: $((x1,y1)×(x2,y2))×(x3,y3)=(x1,y1)×((x2,y2)×(x3,y3))$. Произведения первой со второй и второй с третьей точек салатовые. Итоговые произведения оранжевые, причем совпадают, что и было нужно.

**Текст программы:**

Clear[x,y,f,g,b1,b2,t1,t2,t3,to,x1,x2,x3,xo,y1,y2,y3,yo,x12,

y12,xm12,ym12,x23,y23,xm23,ym23,S12,Sm12,S23,Sm23,

S1m23,S12m3,x12m3,y12m3,x1m23,y1m23]

Manipulate[

f=x^3+b1\*x+b2-y^2; (\*уравнение сечения\*)

g=Sqrt[x^3+b1\*x+b2]; (\*разрешим его относительно y\*)

y1=t1\*g/.x-> x1;

y2=t2\*g/.x-> x2;

y3=t3\*g/.x-> x3; (\*три изначальные точки\*)

yo=to\*g/.x-> xo; (\*единица\*)

(\*перемножаем в порядке ((x1,y1)\*(x2,y2))\*(x3,y3), т.е. проводим прямые, выбираем точки пересечения.\*)

Sm12={x,y}/.NSolve[{f==0,(y-y1)\*(x2-x1)==(x-x1)\*(y2-y1)},{x,y}];

Do[ If [Sm12[[i]][[1]] != x1 && Sm12[[i]][[1]] != x2, {xm12 = Sm12[[i]][[1]], ym12 = Sm12[[i]][[2]]}], {i,3}];

S12={x,y}/.NSolve[{f==0,(y-yo)\*(xm12-xo)==(x-xo)\*(ym12-yo)},{x,y}];

Do[ If[S12[[i]][[1]]!=xm12 &&S12[[i]][[1]]!=xo,{x12=S12[[i]][[1]],y12=S12[[i]][[2]]}],{i,3}];

S12m3={x,y}/.NSolve[{f==0,(y-y3)\*(x12-x3)==(x-x3)\*(y12-y3)},{x,y}];

Do[ If[S12m3[[i]][[1]]!=x12 &&S12m3[[i]][[1]]!=x3,{x12m3=S12m3[[i]][[1]],y12m3 = S12m3[[i]][[2]]}],{i,3}];

(\*перемножаем в порядке (x1,y1)\*((x2,y2)\*(x3,y3))\*)

Sm23={x,y}/.NSolve[{f==0,(y-y3)\*(x2-x3)==(x-x3)\*(y2-y3)},{x,y}];

Do[ If[Sm23[[i]][[1]]!=x2 &&Sm23[[i]][[1]]!=x3,{xm23=Sm23[[i]][[1]], ym23 =Sm23[[i]][[2]]}], {i,3}];

S23={x,y}/.NSolve[{f==0,(y-yo)\*(xm23-xo)==(x-xo)\*(ym23-yo)},{x,y}];

Do[ If[S23[[i]][[1]]!=xm23 &&S23[[i]][[1]]!=xo,{x23=S23[[i]][[1]],y23=S23[[i]][[2]]}],{i,3}];

S1m23={x,y}/.NSolve[{f==0,(y-y1)\*(x23-x1)==(x-x1)\*(y23-y1)},{x,y}];

Do[ If[S1m23[[i]][[1]]!=x23 &&S1m23[[i]][[1]]!=x1,{x1m23=S1m23[[i]][[1]], y1m23 =S1m23[[i]][[2]]}],{i,3}];

(\*рисуем сечение и прямые\*)

ContourPlot[

{

f==0,

(y-y1)\*(x2-x1)==(x-x1)\*(y2-y1),

(y-yo)\*(xm12-xo)==(x-xo)\*(ym12-yo),

(y-y3)\*(x12-x3)==(x-x3)\*(y12-y3),

(y-y3)\*(x2-x3)==(x-x3)\*(y2-y3),

(y-yo)\*(xm23-xo)==(x-xo)\*(ym23-yo),

(y-y1)\*(x23-x1)==(x-x1)\*(y23-y1)

},{x,-l,l},{y,-l,l},

(\*и отмечаем точки\*)

Epilog->{

PointSize[Large], Point[{x1,y1}], Point[{x2,y2}], Point[{x3,y3}], Green,

Point[{x12,y12}], Point[{x23,y23}], Pink,

Point[{xm12,ym12}], Point[{xm23,ym23}], Red,

Point[{xo,yo}], Orange,

Point[{x12m3,y12m3}], Point[{x1m23,y1m23}]

}

],

{{b1,-10},-20,20},{{b2,12},-20,20},{{x1,-3},-10,10},{{x2,0.3},-10,10},{{x3,5},-10,10},{{xo,-2},-10,10},{t1,{1,-1}},{{t2,-1},{1,-1}},{t3,{1,-1}},{{to,-1},{1,-1}},{{l,20},1,100}]



#### Работа с пакетом PolyhedronData и линейное программирование

Формулировка задачи

1. Выбрать выпуклый многогранник.

2. Выбрать одну из его граней.

3. Найти вершину многогранника, наиболее удаленную от выбранной грани как вершину максимизирующую проекцию на внутреннюю нормаль грани.

4. Визуализировать построение.

Программа

(\* 1. Выбор выпуклого многогранника PolyhedronData["Convex"] возвращает список всех выпуклых многогранников. При помощи команды Manipulate добавляем возможность выбора многогранника из раскрывающегося списка. \*)

Manipulate[

(\* 2. Выбор грани После того как выбран многогранник (ph), с помощью PolyhedronData[ph, "FaceCount"] находим количество его граней и с помощью еще одного Manipulate добавляем возможность выбора номера грани (num) из раскрывающегося списка.

3. Поиск наиболее удаленной вершины

PolyhedronData[ph,"VertexCoordinates"] дает список координат вершин, а PolyhedronData[ph,"FaceIndices"] - список граней (каждый элемент этого списка - список из номеров вершин, принадлежащих очередной грани). Сделаем из этих двух списков список граней, где вместо номеров вершин будут стоять их координаты. \*)

fc=Map[(N[PolyhedronData[ph,"VertexCoordinates"]][[#]])&,PolyhedronData[ph,"FaceIndices"]];

(\* Теперь составляем список внутренних нормалей к граням. Для этого у каждой грани берутся три первых вершины (v1, v2, v3) и вычисляется векторное произведение векторов (v3-v1) и (v2-v1). В списке вершин граней, возвращаемом PolyhedronData[ph,"FaceIndices"], номера вершин каждой грани идут в таком порядке, что полученная нормаль всегда оказывается именно внутренней, а не внешней. Полученные нормали не обязаны иметь единичную длину, но для дальнейших вычислений это не имеет значения. \*)

m=Map[(Cross[#[[3]]-#[[1]],#[[2]]-#[[1]]])&,fc];

(\* Чтобы можно было использовать пакет LinearProgramming, надо сначала задать многогранник (вместе с его внутренностью) как пересечение полупространств, то есть при помощи линейных неравенств вида Subscript[A, 1]x+Subscript[A, 2]y+Subscript[A, 3]z>=В. Левые части этих неравенств получить просто: (Subscript[A, 1], Subscript[A, 2], Subscript[A, 3]) - это вектор нормали к грани, найденный выше. Так как равенство достигается в точках плоскости, содержащей соответствующую грань, для нахождения очередного коэффициента В достаточно подставить в уже известную левую часть неравенства координаты первой вершины грани. Подстановка реализуется как скалярное произведение вектора нормали и радиус-вектора вершины. Все найденные коэффициенты В помещаются в список b. \*)

b=MapThread[(#1 . #2[[1]])&,{m,fc}];

(\* Теперь применим функцию LinearProgramming. Она ищет точку многогранника, в которой достигается минимум скалярного произведения радиус-вектора этой точки и внешней нормали к выбранной грани. Условие принадлежности точки x многограннику (или его внутренности) имеет вид (m[[i]], x)>=b для всех i=1, 2, ... n (количество граней многогранника). Найденная точка не обязана быть вершиной многогранника (например, если у многогранника есть параллельные грани, и одна из них выбрана). \*)

Manipulate[res=LinearProgramming[-m[[num]],m,b,-Infinity];

(\* Теперь сделаем так, чтобы показываемая точка была именно вершиной многогранника. Для этого будем последовательно проверять принадлежность найденной точки граням многогранника и добавлять номера тех граней, которым она принадлежит, в новый список (fn). Затем при помощи списка fc найдем общие вершины всех таких граней и заменим найденную точку на одну из них (newres). Если исходная точка уже была вершиной (то есть точкой пересечения трех или более граней), с ней ничего не случится: ведь в этом случае это единственная общая вершина этих граней. Если исходная точка лежала на ребре (то есть на пересечении двух граней), она будет заменена на один из концов этого ребра. Если же исходная точка лежала внутри одной из граней, то она будет заменена на одну из вершин этой грани. \*)

fn=Pick[Range[Length[b]],Map[#== 0&,Chop[m.res-b]]];

newres=(Intersection@@fc[[fn]])[[1]];

(\* 4. Визуализация

При помощи функции Show изображаем на одном рисунке многогранник с выделенной гранью и найденную вершину.Выбранная грань изображается как многоугольник, вершины которого совпадают с вершинами этой грани. Многогранник можно изобразить с помощью функции Graphics3D и пакета PolyhedronData указанным ниже способом.Вместо точки рисуем маленькую сферу:так получается красивее,чем если рисовать именно точку. Синяя точка - результат, полученный функцией LinearProgramming, красная - вершина исходного многогранника, находящаяся на том же расстоянии от заданной грани. \*)

Show[Graphics3D[{Opacity[.6], Glow[], EdgeForm[Gray], PolyhedronData[ph,"Faces"], Opacity[0.9], Green,Polygon[fc[[num]]]}, Lighting->"Neutral"], Graphics3D[{PointSize[Large], Blue, Sphere[res,0.05], Red, Sphere[newres,0.05]}]], {num, 1, PolyhedronData[ph, "FaceCount"], 1, ControlType->PopupMenu}], {ph,PolyhedronData["Convex"]}, ControlPlacement->Top]

Ниже представлены результаты для многогранников: Antiprism4, AugmentedDodecahedron.

 

#### Описание абстрактных операций и пакет FiniteGroupData

Формулировка задачи:

Используя данные описания конечных групп в пакете FiniteGroupData о таблице умножения найти перечень всех нормальных подгрупп в группе $S\_{4}$.

Текст программы:

Clear[mult,n,a,i,j,k,h,u,t,l,sopr,so,nor,flag,fl,norm];

n=FiniteGroupData["Tetrahedral","Order"];

mult=FiniteGroupData["Tetrahedral","MultiplicationTable"];

a=FiniteGroupData["Tetrahedral","Inverses"];

h=Table[0,{q,n},{p,n+1}]; (\*таблица для хранения классов сопряженности; в n+1 столбце находится число элементов класса\*)

byl=Table[0,{i,n}]; (\*таблица для проверки, все ли элементы вошли в какой-нибудь класс сопряженности\*)

sopr=0; (\*счетчик классов сопряженности\*)

For[k=1,k<=n,k++,

l=1;

If[byl[[k]]==1,Continue[],sopr++];

For[j=1,j<=n,j++,

 h[[sopr,l]]=mult[[mult[[j,k]],a[[j]]]]; (\*вычисление сопряженного элемента\*)

 t=0;

 For[i=1,i<l,i++,

 If[h[[sopr,i]]==h[[sopr,l]],h[[sopr,l]]=0;t++] (\*проверка, встречался ли ранее\*)

 ];

 If[j==n,h[[sopr,n+1]]=l-1];

 If[t!=0,Continue[],l++]

 ];

For[i=1,i<=h[[sopr,n+1]],i++,byl[[h[[sopr,i]]]]=1];

]

u=Table[0,{i,sopr-1}]; (\*составляем группу из классов сопряженности; 1-класс принадлежит группе, 0-нет. единичный всегда принадлежит. перебор по всем вариантам\*)

so=2^(sopr-1); (\*количество вариантов\*)

nor=Table[If[i==1||i==n+1,1,0],{i,n+1}]; (\*шаблон для группы\*)

Print["normal`nu:"];

For[i=0,i< so,i++,

For[j=1,j<=4,j++, If[Mod[i,2^j]>=2^(j-1),u[[j]]=1;nor[[n+1]]+=h[[j+1,n+1]],u[[j]]=0]]; (\*собственно перебор\*)

If [Mod[n,nor[[n+1]]]!=0,nor[[n+1]]=1;Continue[]]; (\*отсеивание явных "не групп"\*)

l=2;

For[j=1,j<=4,j++,

If[u[[j]]==1,

 For[k=1,k<=h[[j+1,n+1]],k++, nor[[l]]=h[[j+1,k]]; l++ ] (\*заполнили шаблон\*)

]

];

flag=1;

For[j=2,j<=h[[j+1,n+1]],j++,

 fl=0;

 For[k=2,k<=h[[j+1,n+1]],k++, If[a[[nor[[j]]]]==nor[[k]],fl=1;Break[]]]; (\*проверка, для всех ли найдётся обратный\*)

 If[fl==0,flag=0;Break[]];

];

If[i!=0,If[flag==0,Continue[]] ];

norm=Table[nor[[i]],{i,nor[[n+1]]}]; (\*убираем лишние нули\*)

Print[norm]

]

Результат

normal`nu:

{1}

{1,8,17,24}

{1,4,5,16,21,12,20,9,13,8,17,24}

{1,2,6,3,22,15,7,4,5,16,21,12,20,9,13,8,17,24,10,11,18,14,23,19}

#### Восстановление кривой по кривизне и кручению

Формулировка задачи:

Построить пространственную кривую по её кривизне и кручению, записанных в натуральном параметре.

(\*Для это заводим свободный вектор \*)

$k[t\\_]=1+Sin[t]^{2};κ[t\\_]=Cos[t];γ[t\\_]:=\{x[t],y[t],z[t]\};$

$eq[ξ\\_,η\\_]:=Table[ξ[\left[i\right]]==η[\left[i\right]],\{i,3\}];$;

(\*это векторы скорости, ускорения и бинормали\*)

$v\left[t\_{-}\right]:=\left\{v1\left[t\right],v2\left[t\right],v3\left[t\right]\right\}; n\left[t\_{-}\right]:=\left\{n1\left[t\right],n2\left[t\right],n3\left[t\right]\right\};b[t\\_]:=\{b1[t],b2[t],b3[t]\};$

(\* Запишем уравнение Френе в натуральном параметре. \*)

$УравненияФрене=eq[γ'[t],v[t]]\~Join\~eq[v'[t],k[t]n[t]]\~Join\~eq[n'[t],-k[t]v[t]+κ[t]b[t]]\~Join\~eq[b'[t],-κ[t]n[t]];$

(\*ограничимся численным решением до 100\*)

Решение= NDSolve[УравненияФрене ~Join ~eq[$γ$[0],{0,0,0}] ~Join ~eq[$v$[0],{1,0,0}] ~Join~eq[*n*[0],{0,1,0}] ~Join ~eq[b[0],{0,0,1}], $γ$[t] ~Join ~$ v$[t] ~Join ~n[t] ~Join ~b[t],{t,0,100}];

ParametricPlot3D[$γ$[t]/.Решение, {t,0,100}]

 (\* Для построения бинормали заменим аргумент функции\*)

ParametricPlot3D[$b$[t]/.Решение, {t,0,100}]

Clear[k, $κ$,$ γ$,eq,v,n,b, УравненияФрене, Решение]

 

### Персональные задачи

#### Критерии диагонализуемости и полугамильтоновости систем гидродинамического типа

Дано одномерное уравнение гидродинамического типа

$$(u^{i})\_{t}=A\_{j}^{i}(u^{j})\_{x};i,j=1,...,n,гдеu=(u^{1},...,u^{n})=(u[1],...,u[n]),(A\_{j}^{i})=a=a(u).$$

Если диагональная система полугамильтонова, то она интегрируется "обобщенным методом годографа" Царёва.

Далее приводится вычислительная программа, проверяющая является ли система диагонализуемой и, если является, полугамильтонова ли (при условии различных собственных значений). За диагонализуемость отвечает тождественное равенство нулю компонент тензора Хантьеса, а за полугамильтоновость - тождественное равенство нулю компонент тензора гидродинамической интегрируемости, выведенного в работе "Инвариантный критерий гидродинамической интегрируемости" Павлова, Свинолупова, Шарипова. (все остальные тензоры, функции и векторы вспомогательные!)

Программа

Clear[n,u,a,f0,f1,Nei,Han,t,ei,l,i,j,b,fk,KK,M,fq,fa,Q,fp,P,fla]

n=4;

Array[u,n];a=Array[am,{n,n}];fun=Array[f[u[1],u[2],u[3], u[4]],n];

For[i=1,i<=n,i++,For[j=1,j<=n,j++, If[i==j,a[[i,i]]=fun[[i]],a[[i,j]]=0]]]

(\*задание матрицы А\*)

f0[x\_,y\_,z\_]:=Sum[a[[p,y]]\*D[a[[x,z]],u[p]],{p,1,n}];

f1[x\_,y\_,z\_]:=Sum[a[[y,p]]\*D[a[[p,z]],u[x]],{p,1,n}];

Nei=Table[Simplify[f0[i,j,k]-f0[i,k,j]-f1[j,i,k]+f1[k,i,j]],{i,n},{j,n},{k,n}];

(\*Компоненты тензора Нейенхейса\*)

Han=Table[Simplify[Sum[Nei[[i,p,q]]\*a[[p,j]]\*a[[q,k]]- Nei[[p,j,q]]\*a[[i,p]]\*a[[q,k]]-Nei[[p,q,k]]\*a[[i,p]]\*a[[q,j]]+Nei[[p,j,k]]\*a[[i,q]]\*a[[q,p]],{p,1,n},{q,1,n}]],{i,n},{j,n},{k,n}];

(\*Компоненты тензора Хантьеса\*)

t=0; (\*счетчик количества нулевых компонент тензора Хантьеса\*)

For[i=1,i<=n,i++,For[j=1,j<=n,j++,For[k=1,k<=n,k++,If[Han[[i,j,k]]==0,t++] ]]];

If[t!=n^3,Print["система не диагонализуема"],

Print["диагонализуема"];

(\*различны ли собственные значения?\*)

ei=Eigenvalues[a];

l=0;

For[i=1,i<=n,i++,

For[j=i+1,j<=n,j++,

If[(ei[[i]])==(ei[[j]]), {l=1,Break[]}];If[l==1,Break[]]

]];

If[l==1, Print["есть одинаковые собственные значения"],

Print["собственные значения различны"];

b=a.a;

fk[x\_,y\_,z\_,w\_]:=Sum[(b[[w,p]]\*D[Nei[[p,x,y]],u[z]]-b[[p,z]]\*D[Nei[[w,x,y]],u[p]]+Nei[[p,x,y]]\*D[b[[w,z]],u[p]]-Nei[[w,z,p]]\*D[b[[p,y]],u[x]]+Nei[[w,z,p]]\*D[b[[p,x]],u[y]]),{p,1,n}];

KK=Table[Simplify[fk[i,j,k,s]+fk[j,k,i,s]+fk[k,i,j,s]] ,{s,n},{k,n},{i,n},{j,n}];

M=Table[Simplify[Sum[Sum[Nei[[s,k,p]]\*a[[p,q]]\*Nei[[q,i,j]]+Nei[[s,p,q]]\*a[[p,k]]\*Nei[[q,i,j]]-Nei[[s,p,q]]\*Nei[[p,i,k]]\*a[[q,j]]-Nei[[s,p,q]]\*Nei[[p,k,j]]\*a[[q,i]]-

Nei[[s,k,p]]\*Nei[[p,i,q]]\*a[[q,j]]-Nei[[s,k,p]]\*Nei[[p,q,j]]\*a[[q,i]] ,{q,1,n}],{p,1,n}]],{s,n},{k,n},{i,n},{j,n}];

fq[x1\_,x2\_,x3\_,x4\_,x5\_,x6\_,x7\_,x8\_]:=a[[x1,x2]]\*KK[[x3,x4,x5,x6]]\*a[[x7,x8]];

fa[x1\_,x2\_,x3\_,x4\_,x5\_,x6\_]:=a[[x1,x2]]\*M[[x3,x4,x5,x6]];

Q=Table[ Simplify[Sum[Sum[fq[p,k,s,p,q,j,q,i]+fq[p,k,s,p,i,q,q,j]-fq[p,q,s,p,i,j,q,k]- fq[p,i,s,k,p,q,q,j],{q,1,n}],{p,1,n}]+Sum[4\*fa[p,k,s,p,i,j]-2\*fa[p,i,s,k,p,j]-2\*fa[p,j,s,k,i,p],{p,1,n}]], {s,n},{k,n},{i,n},{j,n}]; fp[x1\_,x2\_,x3\_,x4\_,x5\_,x6\_,x7\_,x8\_]:=a[[x1,x2]]\*Q[[x3,x4,x5,x6]]\*a[[x7,x8]];

P=Table[ Simplify[Sum[Sum[fp[s,p,p,k,q,j,q,i]+fp[s,p,p,k,i,q,q,j]-fp[s,q,p,k,i,j,q,p]-fp[p,i,s,k,p,q,q,j],{q,1,n}],{p,1,n}]] ,{s,n},{k,n},{i,n},{j,n}];

(\*тензор гидродинамической интегрируемости\*)

fla=0;

(\*счетчик количества нулевых компонент\*)

For[i=1,i<=n,i++,

For[j=1,j<=n,j++,

For[k=1,k<=n,k++,

For[s=1,s<=n,s++,If[P[[i,j,k,s]]==0,fla++] ]]]];

If[fla==n^4, Print["система полугамильтонова"], Print["система не полугамильтонова, нулевых компонент ", fla ," из ",n^4]]] ]

Примеры диагонализуемых и полугамильтоновых систем (задание матрицы А)

(\*классическая изотерма Ленгмюра\*)

G=Table[i\*2+8,{i,n}]; (\*произвольные, не равные между собой константы\*)

For[i=1,i<=n,i++,v0[i]=Simplify[G[[i]]\*u[i]/(1+Sum[G[[p]]\*u[p],{p,1,n}])]];

For[i=1,i<=n,i++,For[j=1,j<=n,j++, a[i,j]=Simplify[D[v0[i],u[j]]]

]]

(\*пример из статьи Бялого, Миронова. n=3\*)

For[i=1,i<=n,i++,For[j=1,j<=n,j++, a[i,j]=0]]

a[2,1]=a[3,2]=u[3];

a[1,3]=u[2]; a[2,3]=2\*u[3]-3\*u[1];

a[3,3]=3-2\*u[2];

(\*пример из статьи Бялого, Миронова. n=4\*)

For[i=1,i<=n,i++,For[j=1,j<=n,j++, a[i,j]=0]]

a[2,1]=a[3,2]=a[4,3]=u[4];

a[1,4]=u[2];

a[2,4]=2\*u[3]-4\*u[1];

a[3,4]=3\*u[4]-3\*u[2];

a[4,4]=4-2\*u[3];

(\*система, описывающая квазиклассический предел связанных уравнений Кортевега де Фриза\*)

For[i=1,i<=n,i++,For[j=1,j<=n,j++, If[i==j,a[i,j]=Sum[u[p],{p,1,n}]+2\*u[i],a[i,j]=0 ]]]

#### Вычисление когомологий симплициальных комплексов

#### Постановка задачи:

вычисление гомологий и когомологий симплициального комплекса, заданного перечнем вершин и граней.

(\*Шаг диагонализации матрицы\*)

Step[x0\_,b0\_]:=(y0=x0;r0=b0;For[k0=r0+1,k0<=Length[y0],k0++,{

t0=ExtendedGCD[y0[[r0]][[r0]],y0[[k0]][[r0]]];

z0=Table[Table[ If[y0[[b0]][[b0]]==0,d0=b0;For[l0=b0+1,l0<=Length[y0],l0++,If[y0[[l0]][[b0]]==0,,d0=l0;]];

If[d0==b0,If[j0<b0,y0[[i0]][[j0]],If[j0==Length[y0[[1]]],0,y0[[i0]][[j0+1]]]],

If[i0==b0,y0[[d0]][[j0]],If[i0==d0,y0[[b0]][[j0]],y0[[i0]][[j0]]]]],

If[i0==b0,If[Abs[t0[[1]]]==Abs[y0[[b0]][[b0]]],y0[[i0]][[j0]],y0[[b0]][[j0]]\*t0[[2]][[1]]+y0[[k0]][[j0]]\*t0[[2]][[2]]],

If[i0==k0,If[Abs[t0[[1]]]==Abs[y0[[b0]][[b0]]],y0[[k0]][[j0]]-y0[[b0]][[j0]]\*(y0[[k0]][[b0]]/y0[[b0]][[b0]]),(y0[[b0]][[j0]]\*y0[[k0]][[b0]]+y0[[k0]][[j0]]\*y0[[b0]][[b0]])/t0[[1]]],y0[[i0]][[j0]]]]],{j0,1,Length[y0[[1]]]}],{i0,1,Length[y0]}];y0=z0;} ];y0);

(\*Выдаем на сколько матрица уже диагонализована\*)

IsDiag[x1\_]:=(b1=Length[x1]; For[i1=1,i1<=Length[x1],i1++, For[j1=1,j1<=Length[x1[[1]]], j1++, If[i1==j1,{},{If[x1[[i1]][[j1]]==0, {}, {If[b1>Min[i1,j1], {b1=Min[i1,j1];},{}]}]}]]]; y1=b1;y1);

(\*Диагонализуем матрицу\*)

Diagonalize[x2\_]:=(y2=x2;While[IsDiag[y2]<Max[Length[y2], Length[y2[[1]]]], {z2=Transpose[Step[y2,IsDiag[y2]]];y2=z2;}];y2);

(\*Процедура, которая по заданным дифференциалам выписывает гомологии\*)

DToHom[D\_,Co\_]:=(KerD=NullSpace[Transpose[#]]&/@D;

(\* Размерность симплициального комплекса\*)

DimD=Length[D]+1;

(\*Матрица образа в координатах ядра\*)

ImKer=Join[{ Transpose[D[[1]]]}, Table[LinearSolve[Transpose[KerD[[n]]], Transpose[D[[n+1]]]] , {n,DimD-2}]];

(\*Диагонализуем матрицу из предыдущего действия\*)

PreDiag=Diagonalize[#]&/@ImKer;

Diag=Table[If[Length[PreDiag[[l1]]]==Length[ImKer[[l1]]],PreDiag[[l1]],Transpose[PreDiag[[l1]]]],{l1,1,Length[PreDiag]}];

(\*Свободная часть гомологий\*)

FreePart=Join[Table[k3=0;For[i3=1,i3<=Length[Diag[[l3]]], i3++, If[i3>Length[Diag[[l3]][[1]]], k3++, If[Diag[[l3]][[i3]][[i3]] ==0,k3++, ] ]];k3,{l3,1,Length[Diag]}], {If[ Length[KerD[[DimD-1]]]==0,0, MatrixRank[KerD[[DimD-1]]]]}];

(\*Вычисляем кручения\*)

K0=DeleteCases[#,1]&/@Abs[DeleteCases[Flatten[#]&/@Diag,0,2]];

K1=Flatten[#,1]&/@FactorInteger[K0];

K2=If[Length[#]==0,{},Transpose[#]]&/@K1;

TorsPart=Sort[#]&/@Table[MapThread[Power,K2[[i4]] ],{i4,1,Length[K2]}];

(\*Красиво выводим ответ\*)

K3=Flatten[Table[Drop[#,If[Length[#]==0,0,-1]]&/@ {Partition[Flatten[Join[Table[{"\[DoubleStruckCapitalZ]"," \[CirclePlus] "},{FreePart[[i5]]}],If[i5==Length[FreePart],{},Table[{"\[DoubleStruckCapitalZ]/",TorsPart[[i5]][[j5]],"\[DoubleStruckCapitalZ]"," \[CirclePlus] "}, {j5,1,Length[TorsPart[[i5]]]}]]]],1]}, {i5,1,Length[FreePart]}],1];

If[Co==True,K4=Reverse[K3];,K4=K3;];

MapThread[Print,#]&/@Table[Join[{{l5-1},{" - "}},K4[[l5]]],{l5,1,Length[K4]}];);

(\*Пример\*)

RP2={{1,2,3},{2,3,4},{3,4,5},{4,5,6},{5,6,2},{6,2,1},{1,7,3},{3,7,5},{5,7,2},{2,7,4},{4,7,6},{6,7,1}};

MaxSimplices=RP2;

(\*Количество вершин\*)

Ver=Max[Flatten[MaxSimplices]];

(\*Размерность симплициального комплекса\*)

Dim=Max[Length[#]&/@MaxSimplices];

(\*Задаем все возможные симплексы\*)

Simplices=Drop[SplitBy[SortBy[DeleteDuplicates[Sort[#]&/@Flatten[Map[Subsets,MaxSimplices],1]],Length],Length],1];

(\*Задаем дифференциал гомологий\*) DHom=Drop[Map[Function[x,Table[If[Length[Position[Subsets[x,{Length[x]-1}],Simplices[[Length[x]-1]][[n]] ]]==0,0,(-1)^Position[ Subsets[x,{Length[x]-1}],Simplices[[Length[x]-1]][[n]] ][[1]][[1]]],{n,Length[Simplices[[Length[x]-1]]]}]],#]&/@Simplices,1];

(\*Задаем дифференциал когомологий\*)

DCoHom=Reverse[Transpose[#]&/@DHom];

(\*Выводим ответ на экран\*)

Print["Homology"];

DToHom[DHom,False];

Print["Cohomology"];

DToHom[DCoHom,True];

Результат:

Homology

$$0-Z$$

$$1-Z/2Z$$

$$2-$$

Cohomology

$$0-Z$$

$$1-$$

$$2-Z/2Z$$

#### Нахождение кусочно-линейной аппроксимации кривой, не меняющей ее гомотопический тип

Данная задача достаточно затруднительно формализуется для выполнения в автоматизированной среде моделирования без жестких требований на класс функций, задающих кривую. Поэтому она была сужена до задачи построение аппроксимации, при которой не допускаются «большие» углы между соседними участками ломаной.

Текст программы:

(\*Функции f, g, h задают кривую в пространстве. Они предполагаются гладкими и периодическими с периодом 1. r - радиус-вектор проекции кривой на плоскость. Предполагается, что проекция имеет только трансверсальные самопересечения (хотя на работу программы это не влияет). \*)

f[t\_]:=Sin[8 \[Pi] t]; g[t\_]:=Sin[6\[Pi] t];

 h[t\_]:=0; period=1;

r[t\_]:={f[t],g[t]};

(\*Функция closeQ проверяет, являются ли два вектора "близкими". Для этого проверяем, верно ли, что отношение нормы их разности к норме меньшего из них не превосходит заданного числа vtol. \*)

vtol=0.5; (\*,LessEqual::nord\*)

closeQ[r1\_,r2\_]:=Module[{mnorm=Min[Norm[r1],Norm[r2]]},Quiet[Check[Norm[r1-r2]/mnorm,Infinity,{Power::infy}]<=vtol,{Power::infy}]];

(\*Рекурсивная функция subdivide[t0\_,t1\_,numrec\_] проверяет, являются ли касательные векторы к кривой в точках, соответствующих значениям параметра t0, t1, "близкими" к разности радиус-векторов этих точек. Если это условие не выполнено, делим отрезок [t0, t1] пополам и выполняем проверку для каждой из его половинок. Третий аргумент функции задает максимальную глубину рекурсии. На очередном шаге рекурсии точка t2 добавляется к возвращаемому списку точек разбиения (точек t0 и t1 среди них нет).\*)

empty:=Sequence[]; (\* это нужно, чтобы не добавлять пустые списки \*)

subdivide[t0\_,t1\_,numrec\_]:=If[numrec<=0,empty,Module[{t2=(t0+t1)/2,dt=t1-t0,dr=(r[t1]-r[t0])},If[closeQ[(r'[t0] dt),dr]&&closeQ[(r'[t1] dt),dr],empty,{subdivide[t0,t2,numrec-1],t2,subdivide[t2,t1,numrec-1]}]]]

(\*pts - список точек, полученных при помощи subdivide с выбранной нами глубиной рекурсии\*)

pts=subdivide[0,1,12];

pts//N//TreeForm;

(\*Теперь добавляем к этому списку точки 0 и 1 и делаем список одномерным. Ведь на очередном шаге рекурсии функция subdivide добавляла список точек. Наконец, по построенному списку значений параметра t, соответствующих узлам ломаной, строим список fpts точек на кривой. \*)

pts=Flatten[{0,pts,1}];

fpts=r/@pts;

(\*Изображаем проекцию кривой на плоскость вместе с ломаной, построенной по найденным точкам, и касательными векторами к кривой в узлах ломаной. Длины векторов делятся на константу, кратную длине кривой (чтобы они помещались на рисунке). Они характеризуют скорость движения по кривой. При помощи слайдера можно найти точку кривой, отвечающую данному значению параметра.\*)

clen=NIntegrate[Sqrt[f'[t]^2+g'[t]^2],{t,0,1},PrecisionGoal->2];

Show[Graphics[{Red,Point[fpts],Line[fpts],PointSize[Large],Point[Dynamic[r[qwe]]],Darker[Green,3/4],Arrowheads[Medium], Sequence[(Arrow[{r[#],r[#]+r'[#]/(2clen)}])& /@ pts]}],ParametricPlot[r[t],{t,0,1}]]

Slider[Dynamic[qwe],ImageSize->Large]

Dynamic[qwe]



#### Исследование системы уравнений, описывающей деформации алгебр максимального класса

(\*Для описания деформаций алгебр максимального класса порядка 8 используются следующие 8 уравнений:

-15\*(a4)^2-4\*a2\*a5+3\*a3\*a5+10\*a4\*a5-a3\*a6+2\*a2\*a6+6\*a3\*a4-a4\*a6 = 0;

5\*(a3)^2-4\*a2\*a4-6\*a3\*a4+2\*a2\*a5+a3\*a5 = 0;

6\*(a4)^2-5\*a3\*a5-10\*a4\*a5+5\*a3\*a6+4\*a4\*a6 = 0;

35\*(a5)^2-4\*a2\*a6+2\*a2\*a7+7\*a3\*a5+7\*a3\*a6-3\*a3\*a7-15\*a5\*a6+a5\*a7-28\*a4\*a5 = 0;

-21\*(a5)^2-5\*a3\*a6+5\*a3\*a7+7\*a4\*a5+4\*a4\*a6-3\*a4\*a7+20\*a5\*a6-5\*a5\*a7 = 0;

7\*(a5)^2-6\*a4\*a6+9\*a4\*a7-2\*a4\*a8-15\*a5\*a6+10\*a5\*a7-a5\*a8 = 0;

-70\*(a6)^2-4\*a2\*a7+2\*a2\*a8+11\*a3\*a7-5\*a3\*a8+8\*a3\*a6-44\*a4\*a6+84\*a5\*a6+21\*a7\*a6-a6\*a8-3\*a4\*a7+a4\*a8 = 0;

56\*(a6)^2 -5\*a3\*a7+5\*a3\*a8+9\*a4\*a7 -8\*a4\*a8+8\*a4\*a6 -36\*a5\*a6 -35\*a6\*a7+6\*a6\*a8=0.

Гипотеза 1: Решение этой системы имеет вид:

a2=0,a3=0,a4=0,a5=0,a6=t,a7=4\*t,a8=14\*t;

a2=0,a3=0,a4=0,a5=0,a6=0,a7=t,a8=u;

a2=t,a3=0,a4=0,a5=0,a6=0,a7=0,a8=0;

a2=t/70,a3=t/420,a4=t/2310,a5=t/12012,a6=t/60060,a7=t/291720,a8=t/1385670

Гипотеза 2:Из 8 уравнений одно (8-е) можно выкинуть, то есть множество решений при этом не изменится.

Для проверки гипотезы 2 были вычислены радикалы соответствующих идеалов. Сделано это было с помощью программы Singular (http://www.singular.uni-kl.de).

Радикал для системы уравнений 1-7

j[1]=2\*a3-11\*a4

j[2]=a2\*a8-33\*a4\*a8

j[3]=-35\*a4\*a8+40\*a5\*a7-8\*a5\*a8

j[4]=4\*a4\*a7-19\*a4\*a8

j[5]=4\*a2\*a7-627\*a4\*a8

j[6]=-11405\*a4\*a8-1344\*a5\*a8+14000\*a6^2-4200\*a6\*a7+200\*a6\*a8

j[7]=-3095\*a4\*a8+700\*a5\*a6-56\*a5\*a8

j[8]=14\*a4\*a6-323\*a4\*a8

j[9]=14\*a2\*a6-10659\*a4\*a8

j[10]=-3105\*a4\*a8+140\*a5^2-4\*a5\*a8

j[11]=14\*a4\*a5-1615\*a4\*a8

j[12]=14\*a2\*a5-53295\*a4\*a8

j[13]=7\*a4^2-4199\*a4\*a8

j[14]=7\*a2\*a4-138567\*a4\*a8

Радикал для системы уравнений 1-8

R1=5\*a4 -26\*a5

R2=5\*a3 -143\*a5

R3=5\*a2\*a8 -858\*a5\*a8

R4=28\*a6\*a7 -25\*a5\*a8 -8\*a6\*a8

R5=4\*a5\*a7 -19\*a5\*a8

R6=10\*a2\*a7 -8151\*a5\*a8

R7=70\*a6^2 -322\*a5\*a8 -5\*a6\*a8

R8=14\*a5\*a6 -323\*a5\*a8

R9=35\*a2\*a6 -138567\*a5\*a8

R10=14\*a5^2 -1615\*a5\*a8

R11=7\*a2\*a5 -138567\*a5\*a8 \*)

(\*Решение системы уравнений 1-7 (решение системы j[1] -j [14])\*)

Solve[{2\*a3-11\*a4==0,a2\*a8-33\*a4\*a8==0,-35\*a4\*a8+40\*a5\*a7-8\*a5\*a8==0,4\*a4\*a7-19\*a4\*a8==0,4\*a2\*a7-627\*a4\*a8==0,-11405\*a4\*a8-1344\*a5\*a8+14000\*a6^2-4200\*a6\*a7+200\*a6\*a8==0,-3095\*a4\*a8+700\*a5\*a6-56\*a5\*a8==0,14\*a4\*a6-323\*a4\*a8==0,14\*a2\*a6-10659\*a4\*a8==0,-3105\*a4\*a8+140\*a5^2-4\*a5\*a8==0,14\*a4\*a5-1615\*a4\*a8==0,14\*a2\*a5-53295\*a4\*a8==0,7\*a4^2-4199\*a4\*a8==0,7\*a2\*a4-138567\*a4\*a8==0},{a2,a3,a4,a5,a6,a7,a8}]

{{a3->0,a2->0,a6->0,a4->0,a5->0},{a3->0,a7->1/21 (70 a6+a8),a2->0,a4->0,a5->0},{a3->0,a2->0,a7->(10 a6)/3,a4->0,a5->0,a8->0},{a3->0,a2->0,a7->(10 a6)/3,a4->0,a5->0,a8->0},{a3->0,a2->0,a7->(10 a6)/3,a4->0,a5->0,a8->0},{a3->0,a2->0,a7->(10 a6)/3,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->(46189 a8)/14,a7->(19 a8)/4,a2->(138567 a8)/7,a6->(323 a8)/14,a4->(4199 a8)/7,a5->(1615 a8)/14},{a3->0,a2->0,a7->(19 a8)/4,a6->(79 a8)/56,a4->0,a5->0},{a3->0,a7->a8/5,a2->0,a6->(2 a8)/25,a4->0,a5->a8/35},{a3->0,a7->(1616 a8)/21,a2->0,a6->(323 a8)/14,a4->0,a5->0}}

(\*Решение для системы уравнений 1-8 (решение системы R1 - R11)\*)

Solve [{5\*a4 -26\*a5 == 0,5\*a3 -143\*a5 == 0,5\*a2\*a8 -858\*a5\*a8 == 0,

28\*a6\*a7 -25\*a5\*a8 -8\*a6\*a8 == 0, 4\*a5\*a7 -19\*a5\*a8 == 0, 10\*a2\*a7 -8151\*a5\*a8 == 0, 70\*a6^2 -322\*a5\*a8 -5\*a6\*a8 == 0,14\*a5\*a6 -323\*a5\*a8 == 0, 35\*a2\*a6 -138567\*a5\*a8 == 0, 14\*a5^2 -1615\*a5\*a8 == 0, 7\*a2\*a5 -138567\*a5\*a8 == 0}, {a2, a3, a4, a5, a6, a7, a8}]

(\*Последнее вычисление опровергает гипотезу 2.\*)

{{a3->0,a2->0,a6->0,a4->0,a5->0},{a3->0,a7->1/21 (70 a6+a8),a2->0,a4->0,a5->0},{a3->0,a2->0,a7->(10 a6)/3,a4->0,a5->0,a8->0},{a3->0,a2->0,a7->(10 a6)/3,a4->0,a5->0,a8->0},{a3->0,a2->0,a7->(10 a6)/3,a4->0,a5->0,a8->0},{a3->0,a2->0,a7->(10 a6)/3,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->0,a6->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->0,a7->(10 a6)/3,a2->0,a4->0,a5->0,a8->0},{a3->(46189 a8)/14,a7->(19 a8)/4,a2->(138567 a8)/7,a6->(323 a8)/14,a4->(4199 a8)/7,a5->(1615 a8)/14},{a3->0,a2->0,a7->(19 a8)/4,a6->(79 a8)/56,a4->0,a5->0},{a3->0,a7->a8/5,a2->0,a6->(2 a8)/25,a4->0,a5->a8/35},{a3->0,a7->(1616 a8)/21,a2->0,a6->(323 a8)/14,a4->0,a5->0}}

#### Вычисление мульти-индексного ряда Пуанкаре особенностей

Постановка задачи:

Подсчёт ряда Пуанкаре для факторкольца O(C^2, 0)/(f) в индуцированной диаграммой Ньютона градуировке для заданной функции f с заданной точностью.

Используемые средства: программа написана на C++.

Замечания:

1. Определение ряда Пуанкаре, выражения его коэффициентов через {dim J(k+bin)/J(k+1)}, нормирований на кольце голоморфных функций в C^2, индуцированных f, затем функций порядка на указанном выше факторкольце можно найти в статье [0].
2. Данную работу можно считать численной проверкой утверждения 1 работы [0].
3. На значения коэффициентов ряда Пуанкаре до k-го порядка влияют только мономы разложения f, имеющие степень не больше 2\*k, поэтому программа может считать ряд не только для полиномов.

Идеи и краткое описание алгоритма.

Рассмотрим фильтрацию подпространствами J(l), определяемыми градуировкой v = (v\_i). Задача состоит в том, чтобы на найти размерность подпространства J(k+bin)/J(k+1), где k – неотрицательный целый мультииндекс, bin - строки из 0, 1; 1 - строка из единиц (все строки имеют размер s, определяемый по диаграмме f). В случае бесконечномерного факторпространства полагаем по определению коэффициент ряда раным 0).

Введем обозначения: {u\_i} - заданные f нормирования на кольце O(C^2, 0), {v\_i} - получаемые из них функции порядка: v\_i(g) = max\_{g' = g mod f } u\_i(g').

В дальнейшем будем использовать канонический изоморфизм для векторных пространств U, V, W, где W вложено в U и V:

U/V ~ (U/W)/(V/W). (1)

Используя (1) для подпространств J(k+bin), J(k+1) и M = span<x^p| найдётся i, тч u\_i(x\_p)=k\_i>, получаем, что dim J(k+bin)/J(k+1) = dim [J(k+bin) $∩$ M] / [J(k+1) $∩$ M]. То же можно применить к идеалу (f). Однако по определению M J(k+1) $∩$ M = 0, так что dim [J(k+bin) $∩$ M] / [J(k+1) $∩$ M] = dim J(k+bin) $∩$ M. Таким образом, задача сводится к конечномерному случаю. Осталось перейти к факторкольцу.

Далее, факторпространство J(k+bin) $∩$ M вкладывается в ортогональное дополнение подпространства (f) $∩$ M в O(C^2, 0) относительно скалярного произведения, определяемого на мономах: <x^p, x^q> = {delta}\_p^q. Обозначим это подпространство за J(k+bin)^{ort}, имеющийся базис от (f) $∩$ M - линейные комбинации мономов {a\_i^j\*x^i}. Далее проверим для полученного базиса условия на значение нормирований {u\_i}. Сделаем это сделующим образом.

Рассмотрим сумму:

a\_i^j\*({alpha}\_i\_j\*x^i) + b\_i\*(x^i\*f)|M, (2)

где (x^i\*f)|M есть ортогональная проекция x^i\*f на подпространство M. Тогда любой вектор из J(k)^{ort} единственным образом записывается в виде (2) при нулевых коэффициентах b\_i^j. Задача состоит в том, чтобы получить соотношения на b\_i, при которых в указанной сумме остаются мономы с нормированиями не меньше k+bin, то есть получить систему неодонородных линейных уравнений с неоднородными частями вида линейных комбинаций a\_i^j (3). Она получается одной из двух естественных группировок слагаемых в сумме - по степеням мономов; другая - по номерам переменных. При решении (3) на a\_i^j тоже могут возникнуть соотношения, при требовании которых получаем из имеющегося базиса новый. При этом естественно смотреть на сумму (2) как на элемент тензорного произведения прямой суммы 2 пространств коэффициентов ({a\_i^j} и {b\_i}), и M, который ограничивается на подпространства, аннулирующие уравнения возникшей системы. Это гарантирует, что, например, при уничтожении мономов размерность пространства решений всей системы не зависит от порядка подстановки решений уравнений.

Однако при приведении системы (3) уравнения вида 0 = {gamma}\_i^j\*a\_^i\_j дают соотношения на a\_^i\_j, при которых в неприведённой системе всех уравнений, возникших ранее, у некоторых может занулиться неоднородная часть. Если же составить систему уравнений с новым соотношением на a\_^i\_j, то такое уравнение не войдёт в систему, т.к. в (2) при рассматриваемом на данном шаге мономе коэффициент - линейная комбинация b\_i. Поэтому при каждом новом соотношении на a\_i^j приходится заново составлять и решать (3), например, методом Жордана. Кроме того, нужно следить за тем, чтобы в неоднородные части уравнений (3) коэффициенты, соответствующие нулевым векторам базиса, входили с коэффициентом 0.

В конце этой части алгоритма получается некоторый базис, который в точности порождает J(k+bin) $∩$ M. Т.о., получена размерность

В результате, имеется алгоритм:

 1) Находим базис подпространства M;

 2) Находим базис J(k+bin)^{ort};

 3) Составляем уравнения на b\_i, соответствующие уничтожению мономов в полученном базисе. При возникновении уравнений на a\_i^j подставляем решения и переходим к 3).

Верность результатов программы проверялась при s = 2, то есть для функций с диаграммой Ньютона, состоящей ровно из 2 компактных компонент, для 3 функций. Результат программы совпал с ответом по формуле утверждения 1 работы [0].

Работа с программой

Ввод данных осуществляется через 3 текстовых файла: f\_coefficients.txt, facets.txt и f\_diag\_monomials.txt.

В f\_coefficients.txt вводится число - размер целочисленной решётки, вершины которой естественно отождествлены с мономами от 2 переменных, - и матрица коэффициентов f, где на позиции (i, j) (слева-направо, сверзу-вниз) коэффициент при x^(i, j). В facets.txt вводится число компактных компонент диаграммы Ньютона f и координаты узлов диаграммы (перечисленных слева-направо, сверху-вниз). В f\_diag\_monomials.txt вводится количество мономов f, принадлежащих диаграмме Ньютона f, и их координаты, в том же порядке.

Запуск: exe - файл.

Ссылки: [0]. "Multi-variable Poincare series associated with Newton diagrams", W.Ebeiling & S.M.Gusein-Zade, "Journal of Singularities", vol.1 (2010), pp. 60-68. <http://arxiv.org/abs/0906.0081v1>

#### Следующая задача