КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Решение задач теоретической механики с применением Wolfram Mathematica
Часть 1
Физико-механический практикум

Учебно-методическое пособие

КАЗАНСКИЙ УНИВЕРСИТЕТ
2016

УДК 531(004)
ББК 22.21(3)
Р47

Печатается по рекомендации учебно-методической комиссии
Института математики и механики им. Н.И. Лобачевского
Казанского (Приволжского) федерального университета
протокол № 1 от 22 сентября 2016 года

Составители:
канд. физ.-мат. наук **О.А. Саченков**,
канд. физ.-мат. наук, доц. **Л.У. Султанов,**
Л.Р. Фахрутдинов

Рецензент –
канд. физ.-мат. наук, доц. **А.А. Саченков,**
канд. физ.-мат. наук, доц. **С.А. Кузнецов**

Р47 **Решение задач теоретической механики с применением Wolfram Mathematica Часть 1. Физико-механический практикум:** учеб.-метод. пособие / О.А. Саченков, Л.У. Султанов, Л.Р. Фахрутдинов. – Казань: Казан. ун-т, 2016. – 27 с.

Данное пособие предназначено для проведения физико-механических практикумов, лабораторных, расчетных и курсовых работ по курсам теоретической механики, теории машин и механизмов и прикладной механики. Представлены краткое описание и ход выполнения кинематических и кинетостатических расчетов в среде Mathematica, предложены вопросы для самостоятельного изучения и примеры заданий.

УДК 531(004)
ББК 22.21(3)

СОДЕРЖАНИЕ

С благодарностью нашему учителю
Юрию Павловичу Артюхину

ВВЕДЕНИЕ В MATHEMATICA

Mathematica является широко используемым программным обеспечением в научных и инженерных расчетах. Одним из важных плюсов указанной системы является возможность работы с символьными выражениями, что позволяет производить не только численные расчеты, но и аналитические. Предполагается, что читатель уже знаком с программным продуктом Mathematica, но рассмотрим некоторые синтаксические и логические аспекты среды. Далее по тексту код в Mathematica будет выделен `шрифтом Courier`, ***курсивом*** будут отмечены задания для самостоятельной работы. Реализация рассмотренных задача приведена в девятой версии среды, но в большей степени будут работоспособны и для более ранних версий.

Все данные в Mathematica заносятся в ячейки (Cells). Для запуска на решение активной ячейки используется сочетание клавиш *Shift+Enter*, для запуска на решение всего фала *Evaluation->Evaluate Notebook* (действия в ячейках выполняется пошагово сверху вниз). Для очистки (обнуления) памяти *Mathematica Evaluation->Quit Kernel->Local*. При обращении к функциям и переменным Mathematica разделяет регистр, то есть *Sin, sin и siN* – три разных имени. Для разделения объектов (переменные, функции) в Mathematica существует инструмент контекста. Внешне контекст задается в виде «*Контекст`*» (обратный апостроф). Каждый объект в Mathematica обладает своим контекстом, который записывается перед именем объекта, обратный апостроф «`» является в этом случае разделителем, то есть в общем виде обращение к переменной выглядит *Контекст`Имя*. По умолчанию имя контекста не отображается. Проиллюстрируем это на примере:

```
h=1
h`h=2
```

Здесь две переменные *h*, но первая создана с контекстом по умолчанию, а вторая создана с контекстом *h`*. Теперь введем следующие команды:

```
h=3
h`h
```

Переменная *h* с контекстом по умолчанию изменила своё значение, а переменная *h`h* нет, так как к ней обращения не было. Для просмотра контекста заданной переменной используется функция *Context[var]*.

Обращение к функциям в Mathematica бывает двух типов. Стандартный тип – *имя_функции[набор_параметров]*, второй тип – постфиксная: *набор_параметров//имя_функции*.

```
Sin[Pi/4]
Pi/4//Sin
```

При обращении к функции возможно задание не только аргументов, но и параметров, которые определяют особенности выполнения функции; для отображения всех опций используется функция *Options[obj]*.

Для удобного переноса формульных результатов из Mathematica в текстовые редакторы удобно использовать функцию *TraditionalForm[expr]*, которая преобразует формулы в традиционный вид. Для копирования в MathType используйте *Copy As->MathML* или *Copy As->LaTeX* для передачи данных в TeX.

```
Sin[x]^2/(1-Log[x])
Sin[x]^2/(1-Log[x])//TraditionalForm
```

Для задания собственных функций используют конструкцию *func[arg_]=expr*, здесь имя функции – *func, arg* – аргумент функции, который должен оканчиваться символом «_», *expr* – выражение, зависящее от *arg*. Например:

```
Sq[a_,b_]=a*b;
```

Здесь задана функция двух аргументов, вычисляющая их произведение, здесь символ «*;*» обозначает что после исполнения команды результат выводиться не будет.

При необходимости получения дополнительной информации о прочих способах использования функций или их опциях рекомендуется использовать уже упомянутую выше команду *Options* или справку Mathematica.

РАБОТА №1. КИНЕМАТИЧЕСКИЙ РАСЧЕТ

1. Основное содержание работы

Задачей является построение кинематических характеристик предложенного механизма в среде Mathematica, получение навыков работы с вектор–функциями, применение к ним таких элементарных действий, как подстановка, дифференцирование. Построение различных типов графиков и анимации.

2. Цель работы

Для заданного механизма определить закон движения, скорости и ускорения точек, построить их траектории.

3. Порядок проведения работы

Рассмотрим порядок проведения работы на примере движения кривошипно-ползунного механизма (КПМ) (рис. 1). Механизм состоит из кривошипа (OA) длиной L_1, соединенного одноподвижными вращательными парами с неподвижной стойкой в точке O и с шатуном AB длиной L_2 в точке A. Шатун AB соединен одноподвижной вращательной парой с ползуном B в точке B. Ползун B, в свою очередь, соединен одноподвижной поступательной парой с неподвижной стойкой.

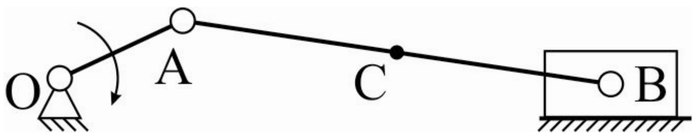


Рис. 1. Кривошипно-ползунный механизм

Длины всех звеньев считаются известными величинами. Точка C – центр масс шатуна и расположена посередине отрезка AB. Механизм обладает одной степенью свободы. Закон вращательного движения начального звена (кривошипа OA) известен – вращение кривошипа с угловой скоростью ω. Необходимо определить закон движения, траектории, скорости и ускорения точек A, B, C.

Для получения закона движения составим векторное уравнение контура механизма. Для этого поместим начало инерциальной системы координат в точку O, ось x направим по линии движения ползуна, а ось y – перпендикулярно к оси x по правилу правой тройки (см. рис. 2). Введем соответствующие звеньям вектора: $\overrightarrow{OA}, \overrightarrow{AB}, \overrightarrow{OB}$.

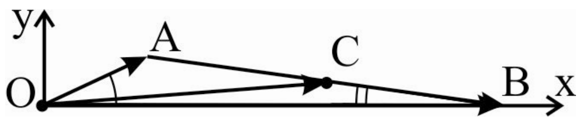


Рис. 2. Векторный контур КПМ

Тогда для любого положения должно выполняться векторное уравнение:

$$\overrightarrow{OA} + \overrightarrow{AB} = \overrightarrow{OB} \qquad (1.1)$$

Введем углы:

$$\begin{aligned} \varphi &= \angle AOB \\ \lambda &= \angle OBA \end{aligned} \qquad (1.2)$$

Тогда компоненты вектора можно расписать:

$$\begin{aligned} \overrightarrow{OB} &= (S, 0) \\ \overrightarrow{OA} &= (l_1 Cos\varphi, l_1 Sin\varphi) \\ \overrightarrow{AB} &= (l_2 Cos\lambda, -l_2 Sin\lambda) \end{aligned} \qquad (1.3)$$

Отсюда получим разрешающую систему:

$$\begin{cases} l_1 Cos\varphi + l_2 Cos\lambda = S \\ l_1 Sin\varphi - l_2 Sin\lambda = 0 \end{cases} \qquad (1.4)$$

Рассмотрим решение указанной задачи в программе Mathematica.

3.1. Построение закона движения

Первый вариант решения

Рассмотрим систему (1.4). Введем переменную *equat* для хранения уравнений системы (1.4), для этого будем использовать символ *&&* – логическое «и». Введем переменные *phi* для хранения φ, lm для хранения λ, *S* – для хранения перемещения ползуна, *L1* и *L2* – для хранения длин звеньев OA и AB соответственно. Запишем систему (1.4):

```
equat=L1*Cos[phi]+L2*Cos[lm]==S&&L1*Sin[phi]-
L2*Sin[lm]==0
```

Далее вызовем функцию *Solve[expr, vars]*, которая решает заданное уравнение *expr* относительно *vars*. Результат решения запишем в переменную *solu*:

```
solu=Solve[equat,{S,lm}]
```

Результатом выполнения функции является вектор решений (или матрица, если решений несколько). Выведем полученные решения в матричном виде:

```
solu//MatrixForm
```

В результате получим следующее выражение:

$$\begin{pmatrix} S \to L1\,Cos[phi] - \sqrt{L2^2 - L1^2\,Sin[phi]^2} & lm \to ConditionalExpression\left[ArcTan\left[-\frac{\sqrt{L2^2 - L1^2\,Sin[phi]^2}}{L2}, \frac{L1\,Sin[phi]}{L2}\right] + 2\pi C[1], C[1] \in Integers\right] \\ S \to L1\,Cos[phi] + \sqrt{L2^2 - L1^2\,Sin[phi]^2} & lm \to ConditionalExpression\left[ArcTan\left[\frac{\sqrt{L2^2 - L1^2\,Sin[phi]^2}}{L2}, \frac{L1\,Sin[phi]}{L2}\right] + 2\pi C[1], C[1] \in Integers\right] \end{pmatrix}$$

Рис. 3. Результат расчета

Таким образом, у нас есть два решения, которые представлены строками матрицы (см. рис. 3). Выражения для угла lm определяются с точностью до периода, в синтаксисе Mathematica это определяется функцией

ConditionalExpression[expr,cond], которая значению функции сопоставляет выражение *expr*, если *cond* верен (*True*). Стоит отметить, что в решении присутствует функция *ArcTan[x,y]* – это просто удобная запись для *ArcTan[y/x]*.

Для вывода каждого решения используется обращение:

```
solu[[1]]
solu[[2]]
```

Конечно, можно было задать уравнения для функции решения в следующем виде:

```
solu=Solve[{L1*Cos[phi]+L2*Cos[lm]==S,L1*Sin[phi]-
L2*Sin[lm]==0},{S,lm}]
```

но при большом количестве уравнений, это уменьшает читабельность кода.

Второй вариант решения

В случае многозвенных механизмов удобнее использовать векторный формализм, что позволяет сохранить читабельность для сложных цепей. Для этого введем вектора $\overrightarrow{OA}, \overrightarrow{AB}, \overrightarrow{OB}$:

```
OA={L1*Cos[phi],L1*Sin[phi]}
AB={L2*Cos[lm],-L2*Sin[lm]}
OB={S,0}
```

Запишем векторное уравнение для функции *Solve*. В переменной *solu2* будем хранить результаты решения, которые совпадают с полученными в первом методе решения.

```
solu2=Solve[OA+AB==OB,{S,lm}]
```

Перейдем к построению положения механизма. Определим вектор значений системы: длины звеньев, значение угла, константа для периодики угла λ. Для этого введем вектор *cnst*:

```
cnst={L1→0.5,L2→2,phi→Pi/3,C[1]→0}
```

Для подстановки в полученные решения используем функцию *ReplaceAll* в постфиксной форме *expr/.rules*. В этом случае в выражении *expr* будет произведена подстановка согласно правилу *rules*. Подставим в вектора $\overrightarrow{OA}, \overrightarrow{AB}, \overrightarrow{OB}$ решения из *solu2* с подстановкой параметров из *cnst*:

```
OA/.solu2[[1]]/.cnst
AB/.solu2[[1]]/.cnst
OB/.solu2[[1]]/.cnst
```

3.2. Построение положений механизма

Для построения положения механизма воспользуемся функций *Line[{{p_{11},p_{12},..},{p_{21},p_{22},..},...}]*, которая определяет ломанную, проходящую через заданные координаты вершин. Для отображения графических данных будем использовать функцию *Graphics[primitives, options]*, которая выводит на экран набор графических примитивов *primitives* (в нашем случае ломанная) с заданными настройками *options*.

```
Graph-
ics[Line[{{0,0},OA/.solu2[[1]]/.cnst,OB/.solu2[[1]]/.cns
t}]]
```

Получим изображение положения:

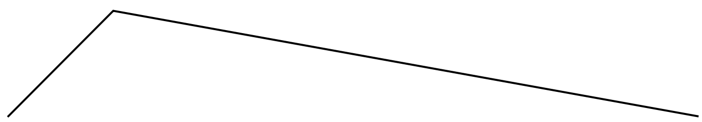


Рис. 4. Положение КПМ

Изменяя параметр *phi* в векторе подстановки *cnst*, получим положения при разных значениях угла φ.

Рассмотрим реализацию анимации движения механизма. Для этого воспользуемся командой *Manipulate[expr,{u,u_{min},u_{max},du}]*, которая возвращает значения *expr* в зависимости от значения параметра *u* в диапазоне *{u_{min},u_{max}}* с шагом *du*. Для этого введем новый вектор параметров *cnst2*, в котором исключим переменную *phi*, а в выражение для построения ломаной добавим конструкцию */.phi->t*. Эта конструкция произведет в выражении замену переменной *phi* на параметр *t* для функции *Manipulate*:

```
cnst2={L1→0.5,L2→2,C[1]→0}
Manipulate[Graphics[Line[{{0,0},OA/.solu2[[2]]/.cnst2/.p
hi→t,OB/.solu2[[2]]/.cnst2/.phi→t}],PlotRange→{{-
1,3},{-1,1}}],{t,0,100,0.5}]
```

Результатом функции *Manipulate* будет окно, приведенное на рис. 5. Окно содержит название изменяемого параметра (пункт 1 на рис. 5), слайдер для изменения значения параметра (пункт 2 на рис. 5), поле вывода (пункт 10 на рис. 5) и разворачиваемое окно (пункт 3 на рис. 5), которое по умолчанию скрыто. При нажатии на 3 (см. рис. 3) появится окно настроек, где показано значение изменяемого параметра (пункт 4 на рис. 5). Также есть кнопки увеличения (пункт 5 на рис. 5) и уменьшения (пункт 6 на рис. 5) на заданный шаг (*du*) параметра, кнопка включения анимации (пункт 7 на рис. 5), кнопки регуляции скорости воспроизведения анимации (пункт 8 на рис. 5) и направления анимации (пункт 9 на рис. 5).

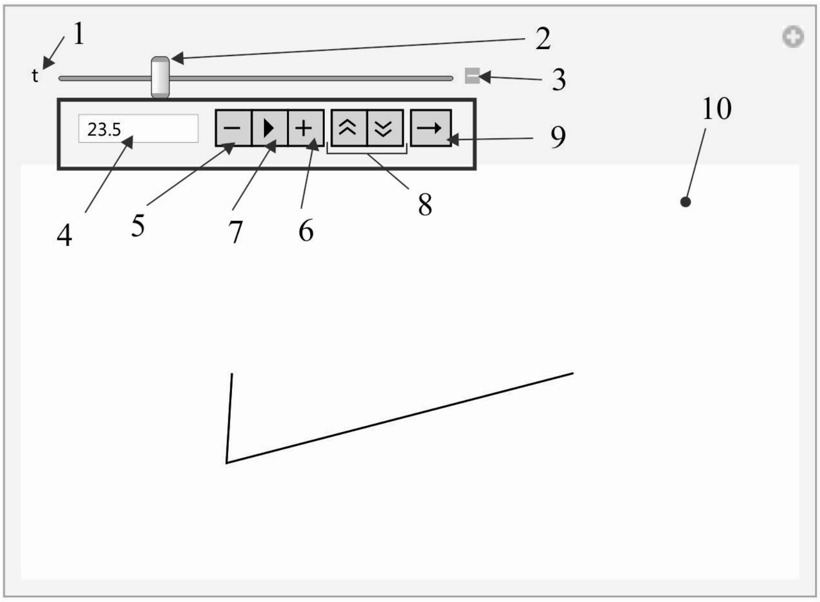


Рис. 5. Окно анимации

На практике удобней хранить полученные решения в виде функций. Для этого определим функции *oa, ab, ob,* аргументом которых является время – t, а параметром – угловая скорость ω (*omega*):

```
oa[t_,omega_]=OA/.solu2[[2]]/.cnst2/.phi→omega*t
ab[t_,omega_]=AB/.solu2[[2]]/.cnst2/.phi→omega*t
ob[t_,omega_]=OB/.solu2[[2]]/.cnst2/.phi→omega*t
```

Задание: *постройте вектор–функцию для oc(t,ω).*

Подставим введенные функции в выражения для анимации:

```
Manipulate[Graphics[Line[{{0,0},oa[t,omega],ob[t,omega]}
],PlotRange→{{-1,3},{-
1,1}}],{t,0,100,0.5},{omega,1,10,1}]
```

3.3.Построение кинематических характеристик

Рассмотрим построение графиков. В Mathematica для этого существует большой набор функций; в случае, когда функция задана в явном виде, удобно использовать функцию *Plot[f,{x,x_{min},x_{max}}, opt]*. Стоит отметить, что при этом все значения функции *f* (кроме аргумента) должны быть заданы численными значениями. Функция *Plot* может быть задана с набором параметров *opt*, которые определяют настройки графического отображения: подписи осей, вид и цвет кривых и т. д. Рассмотрим некоторые из них. В первую очередь это *PlotLegends*, которая определяет подписи к графикам, и *PlotStyle*, которая определяет цвет и тип линии для отображения:

```
Plot[oa[t,1],{t,0,10},PlotLegends->Automatic,PlotStyle-
>{Thick,Dashed}]
```

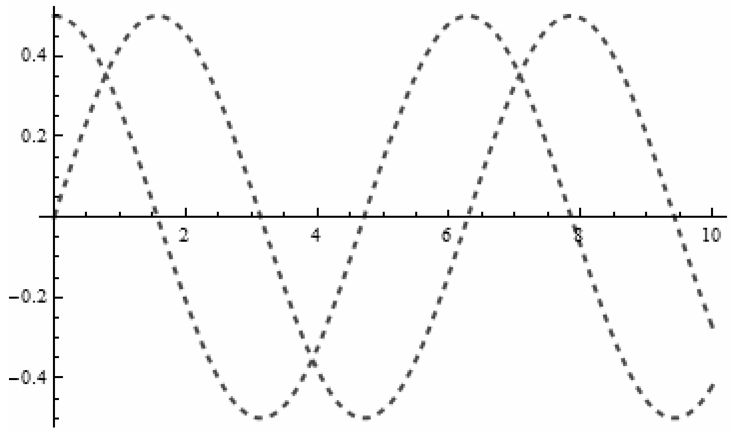


Рис. 6. График вектор-функции закона движения точки А

Как видно из рис. 6, хотя там и отображено два графика – компоненты вектора *oa*, но легенды к графику не появилось, и обе кривых нарисованы однотипно. Функция *Plot* в этом случае воспринимает заданный вектор *oa* как одно цельную функцию. Для того чтобы разделить компоненты вектора, необходимо воспользоваться функцией *Evaluate[expr]*:

```
Plot[Evaluate[oa[t,1]],{t,0,10},PlotLegends→"Expression
s",PlotStyle->{Thick,Dashed}, AxesLabel->{t,f}]
```

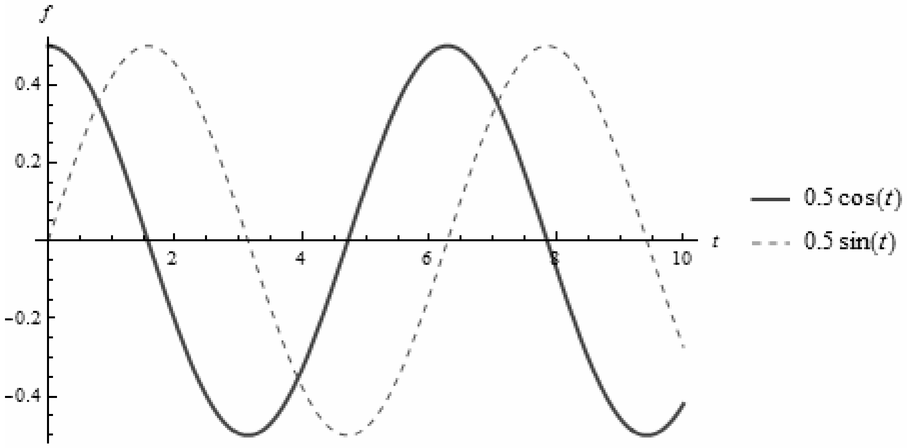


Рис. 7. График вектор-функции закона движения точки А

В этом случае обе компоненты вектора разделены по типу линий, и появилась легенда к графику. Параметр *AxesLabel* определяет подписи осей графика. Значение для параметра *PlotLegends "Expressions"* означает, что в легенде будет отображаться математическое представление заданной функции. Поменяем значение вышеупомянутого параметра:

```
Plot[{oa[t,1][[1]],oa[t,1][[2]]},{t,0,10},PlotLegends→{
"OAx","OAy"}, AxesLabel->{t,f}]
```

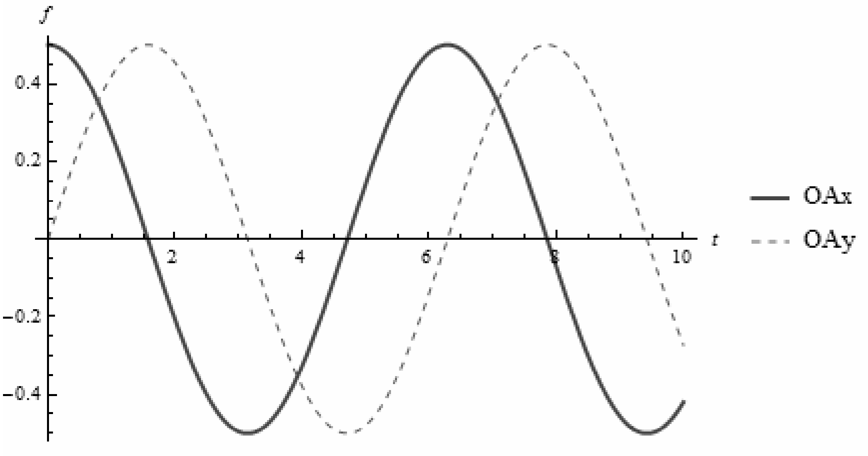


Рис. 8. График вектор-функции закона движения точки А

На практике возникает необходимость выводить семейство функций при различных значениях параметра. Для этого удобно воспользоваться функцией *Table[expr, {i, i_{min}, i_{max}, di}]*, которая возвращает список (вектор) выражения *expr* для всех *i* из диапазона *{i_{min}, i_{max}}* с заданным шагом *di* (по умолчанию 1):

```
Plot[Evaluate[Table[oa[t,p][[1]],{p,1,3,1}]],{t,0,10},PlotLegends→Automatic, AxesLabel->{t,f}]
```

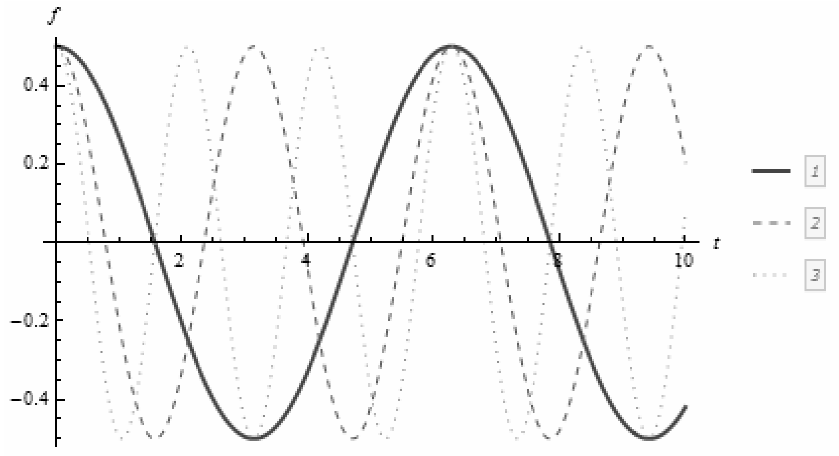


Рис. 9. График вектор-функции закона движения точки А

Для построения траектории необходимо построить график вектор-функции как параметрическую кривую (параметром будет время *t*). Применим для этого функцию *ParametricPlot[f,{x,x_{min},x_{max}}, opt]* и получим траекторию заданной точки. Для начала построим траекторию точки А:

```
ParametricPlot[oa[t,1],{t,0,10}, AxesLabel->{x,y}]
```

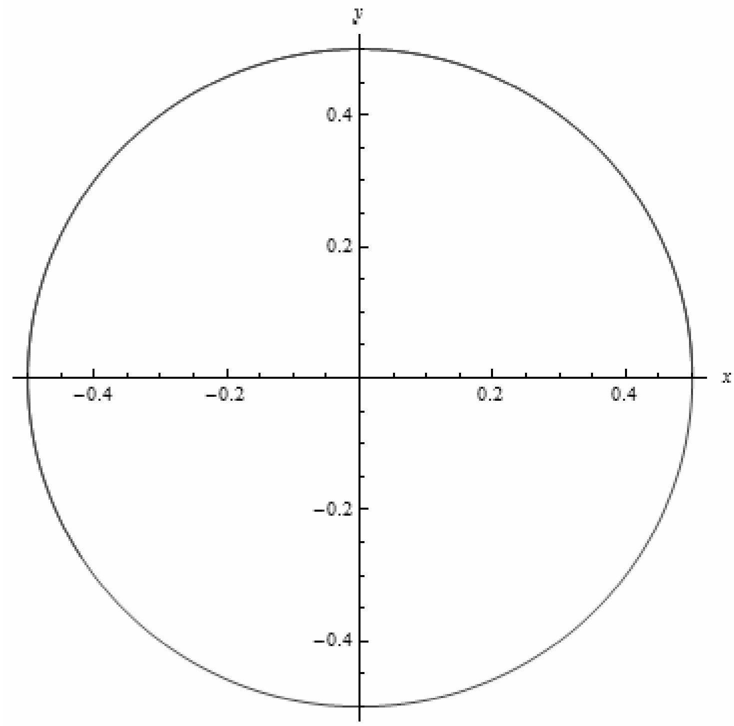


Рис. 10. Траектория точки А

Построим траекторию центра масс второго звена - это точка С на рис. 1:

```
ParametricPlot[oc[t,1],{t,0,10}, AxesLabel->{x,y}]
```

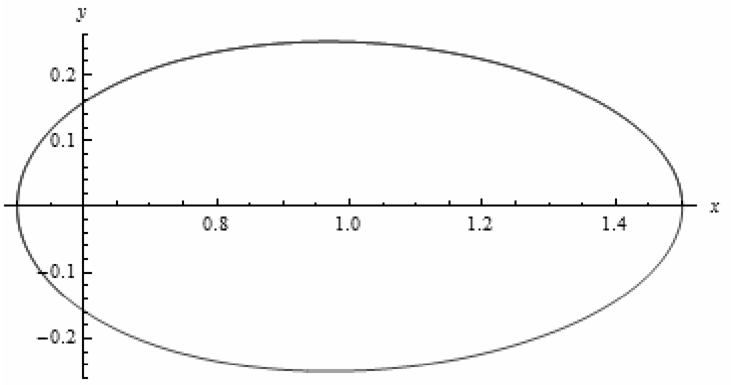


Рис. 11. Траектория точки С

Задание: *постройте траекторию центра масс шатуна с началом координат в точке (0,0).*

Задание: *увеличьте размер шрифта для подписей осей.*

Перейдем к кинематическим характеристикам: скорость и ускорение. По определению скорость – производная от радиус-вектора точки по времени. Для нахождения производной воспользуемся функцией *D[f, x]*, которая определяет производную от функции *f* по *x,* и построим график:

```
Voa[t_,omega_]=D[oa[t,omega],t]
Plot[Evaluate[Voa[t,1]],{t,0,10},PlotLegends→"Expressions", AxesLabel->{t,f}]
```

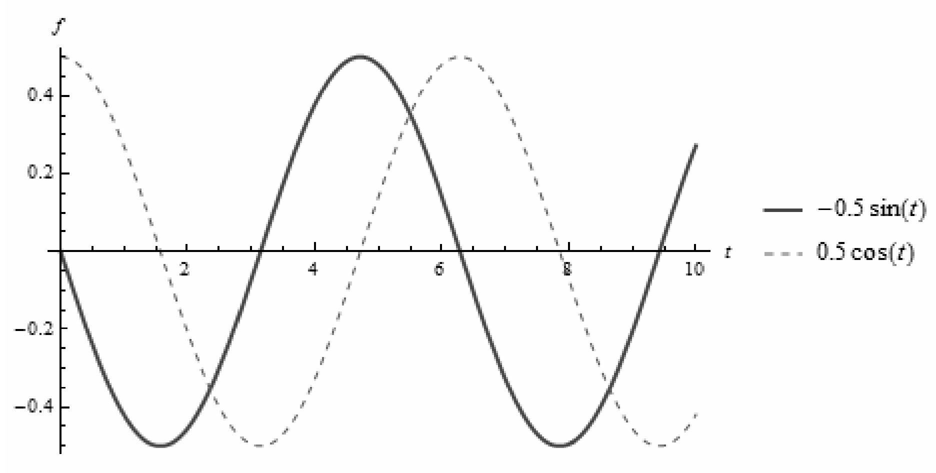


Рис. 12. Скорость точки А

Для нахождения ускорения нам необходимо найти вторую производную от радиус-вектора точки по времени. Для нахождения производной воспользуемся уже знакомой функцией *D[f, {x,n}]*, которая определяет *n*-ю производную от функции *f* по *x*. Построим график:

```
Woa[t_,omega_]=D[oa[t,omega],{t,2}]
Plot[Evaluate[Woa[t,1]],{t,0,10},PlotLegends→"Expressions", AxesLabel->{t,f}]
```

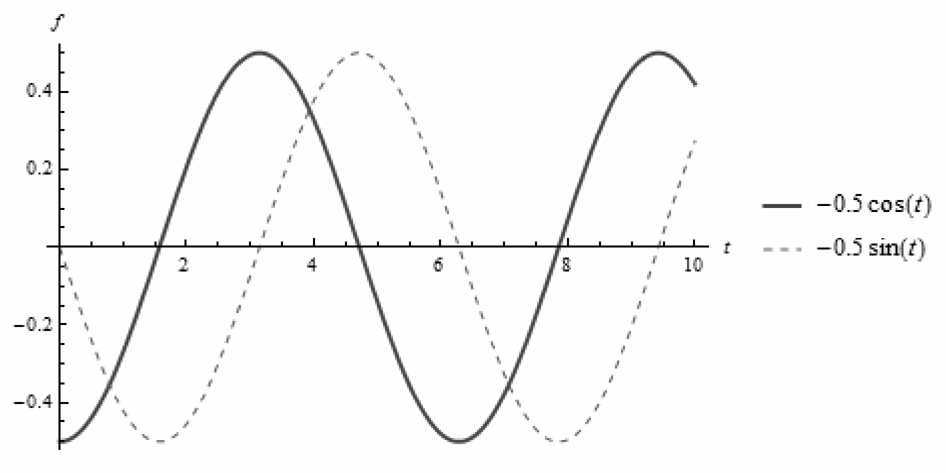


Рис. 13. Ускорение точки А

Задание: *получите скорости и ускорения для точек B и C.*

Чтобы определить величину скорости, воспользуемся функцией *Norm[expr]*, определяющей в нашем случае норму вектора *expr*:

```
Norm[Voa[t,omega]]
Norm[Woa[t,omega]]
```

Перейдем к построению анимации – для этого определим функцию *KPM*, которая будет хранить контур механизма в зависимости от аргумента *t* и параметра *ω*. Также определим функцию *Vc*, которая будет отображать вектор скорости. Упомянутые переменные *KPM* и *Vc* будем использовать с контекстом *G* (от английского Graphics). Для отображения векторов воспользуемся командой *Arrow*, которая однотипна с уже знакомой нам командой *Line* за исключением того, что полученная ломанная линия оканчивается стрелкой:

```
G`KPM[t_,omega_]=Line[{{0,0},oa[t,omega],ob[t,omega]}];
G`Vc[t_,omega_]={Red,Arrow[{oc[t,omega],oc[t,omega]+vc[t
,omega]}]};
Manipulate[Graphics[{G`KPM[t,omega],
G`Vc[t,omega]},PlotRange->{{-1,3},{-
1,1}}],{t,0,100,0.125},{omega,1,5,1}]
```

Здесь в функции Manipulate задано два аргумента для управления – это время t и угловая скорость ω.

Задание: *постройте анимацию для ускорения центра масс шатуна.*

Задание: *поменяйте закон движения начального звена на равноускоренный и посмотрите результаты.*

Определим направляющие косинусы для звеньев:

```
Noa[t_,omega_]=oa[t,omega]/Norm[oa[t,omega]];
Nab[t_,omega_]=ab[t,omega]/Norm[ab[t,omega]];
Nob[t_,omega_]=ob[t,omega]/Norm[ob[t,omega]];
```

Теперь можно найти проекцию вектора скорости на звено AB, умножив скалярно вектор скорости точки C и единичный вектор направления $\overrightarrow{AB}$; в Mathematica для операции скалярного умножения используется символ «.» (точка). Приведем команду построения анимации проекции вектора скорости на звено AB:

```
G`Vpc[t_,omega_]={Blue,Arrow[{oc
[t,omega],oc[t,omega]+(vc[t,omega].Nab[t,omega])*Nab[t,o
mega]}]};
```

Задание: *постройте анимацию проекции вектора скорости на перпендикуляр к звену AB.*

РАБОТА №2. КИНЕТОСТАТИЧЕСКИЙ РАСЧЕТ

1. Основное содержание работы

Задачей работы являются: построение в среде Mathematica силовых характеристик заданного механизма с учетом его движения; получение навыков работы с векторами решений; работа с контекстами в Mathematica, построение массивов векторов; построение различных типов графиков и анимаций.

2. Цель работы

Для заданного механизма определить реакции в шарнирах, найти уравновешивающую реакцию.

3. Определение реакций связей

Рассмотрим задачу кинетостатики для КПМ. Необходимо: определить реакцию в точках A, O; построить годограф реакции в точке O, построить план сил.

В кинетостатике второй закон Ньютона рассматривается как равновесие внешних и инерциальных сил:

$$\vec{F}_i + \vec{F} = 0\,, \tag{2.1}$$

где $\vec{F}_i$ – силы инерции, а $\vec{F}$ – сумма внешних сил.

Силы инерции по определению равны ускорению взятым с обратным знаком и умноженному на массу. Движение механизма понимается как нахождение положения равновесия всех внешних сил и сил инерции.

Для расчета многозвенных механизмов удобно понимать их как составные конструкции, в этом случае такие конструкции разбиваются на подконструкции. В точках разделения конструкции влияние отделяемой части заменяется реакцией связи. Рассмотрим применение этого метода для решения нашей задачи. В этом случае удобно разделить механизм в точке A, для компенсации влияния оставшейся части в точке A введем реактивную силу $\vec{R}_a$ (R_{ax}, R_{ay}), при этом, если в точке A подконструкции AB действует реакция R_a, то в точке A подконструкции OA действует реакция $-\vec{R}_a$. В точке B введем реакцию опоры $\vec{R}_b$ (0, R_{by}). В точке O введем реакцию опоры $\vec{R}_o$ (R_{ox}, R_{oy}). Обозначим векторами силу тяжести звеньев, а также ускорение центра масс второго звена и соответствующую ему силу инерции (см. рис. 14).

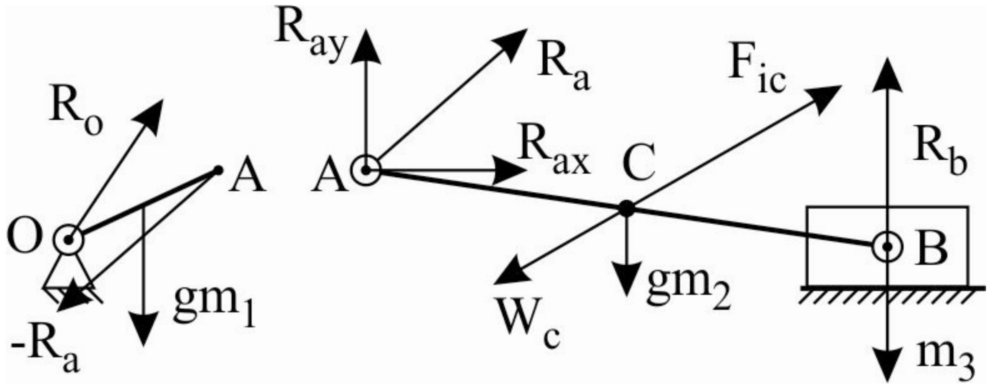


Рис. 14. Расчетная схема

Для составления уравнений моментов необходимо определить плечо от заданной точки до заданного вектора силы:

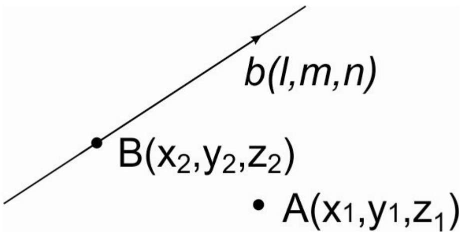


Рис. 15. Определение расстояния от точки до вектора

Из курса аналитической геометрии известно, что расстояние от точки до прямой (направляющий вектор которой известен) можно подсчитать по формуле:

$$h = \frac{|\overrightarrow{AB} \times \vec{b}|}{|\vec{b}|}, \tag{2.2}$$

где $\vec{b}$ – вектор направления прямой, $\overrightarrow{AB}$ – вектор, соединяющий точку A и точку прямой (см. рис. 15). В случае векторов, лежащих в одной плоскости, легко получить:

$$\overrightarrow{AB} = (AB_x, AB_y, 0)$$

$$\vec{b} = (b_x, b_y, 0)$$

$$\overrightarrow{AB} \times \vec{b} = \begin{pmatrix} \vec{i} & \vec{j} & \vec{k} \\ AB_x & AB_y & 0 \\ b_x & b_y & 0 \end{pmatrix} = (0, 0, AB_x \cdot b_y - AB_y \cdot b_x)$$

$$|\overrightarrow{AB} \times \vec{b}| = AB_x \cdot b_y - AB_y \cdot b_x$$

$$\frac{|\overrightarrow{AB} \times \vec{b}|}{|\vec{b}|} = \frac{AB_x \cdot b_y - AB_y \cdot b_x}{\sqrt{b_x^2 + b_y^2}} \tag{2.3}$$

Определим функцию, подсчитывающую плечи векторов:

```
Dist[a_,b_]=(a[[1]]*b[[2]]-a[[2]]*b[[1]])/Norm[b];
```

Зададим неизвестные реакции и массовые параметры системы:

```
Ro={Rox,Roy};
Ra={Rax,Ray};
```

```
Rb={0,Rby};
cnst3={m1->1,m2->2,m3->2,g->9.8};
mass={0,-1};
```

Составим уравнения статики для подконструкции AB с учетом инерциальных сил и запишем их решения в переменную *soluR1*. Здесь в постфиксной записи используется функция *Simplify[expr]*, которая упрощает алгебраическое выражение *expr* – в нашем случае это полученное решение.

```
soluR1=Solve[{Ra+Rb+m2*g*mass+m3*g*mass-m2*wc[t,omega]=={0,0},Dist[ab[t,omega]*0.5,mass]*m2*g+Dist[ab[t,omega],mass]*m3*g+Dist[ab[t,omega],-mass]*Rby-Dist[ab[t,omega]*0.5,wc[t,omega]]*m2*Norm[wc[t,omega]]==0},{Rax,Ray,Rby}]//Simplify
```

Определим функции решения *f`Ra* и *f`Rb* (контекст *f*) на основании полученного решения *soluR1*:

```
f`Ra[t_,omega_]=Ra/.soluR1[[1]]/.cnst3;
f`Rb[t_,omega_]=Rb/.soluR1[[1]]/.cnst3;
```

Составим уравнения статики для подконструкции OA с учетом инерциальных сил, реакции R_a и запишем их решения в переменную *soluR2*:

```
soluR2=Solve[{Ro-f`Ra[t,omega]+m1*g*mass=={0,0},Dist[oa[t,omega]*0.5,mass]*m1*g-Dist[oa[t,omega],f`Ra[t,omega]]*Norm[f`Ra[t,omega]]==0},{Rox,Roy}]//Simplify
```

Определим функцию решения *Rof* на основании полученного решения *soluR2*:

```
f`Ro[t_,omega_]=Ro/.soluR2[[1]]/.cnst3;
```

Задание: *решите задачу с учетом момента инерции второго звена – шатуна, момент инерции удобно задать как константу J.*

4. Построение уравновешивающей силы и плана сил

Построим закон для реакции R_o и годограф реакции:

```
Plot[Evaluate[f`Ro[t,1]],{t,0,10},PlotLegends->Automatic,PlotStyle->{Thick,Dashed}]
ParametricPlot[Evaluate[f`Ro[t,1]],{t,0,10},PlotLegends->Automatic,AxesLabel->{x,y}]
```

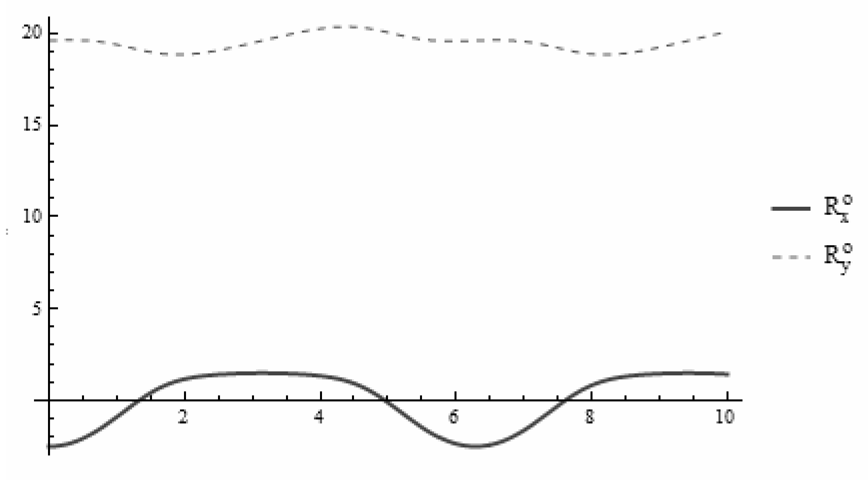


Рис. 16. Реакция в точке О

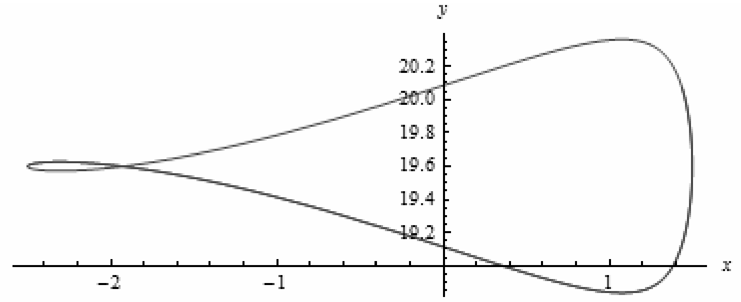


Рис. 17. Годограф вектора реакции в точке О

Построим анимацию механизма со всеми силами. При выводе красными векторами будем обозначать силы тяжести звеньев, синими – реакции, зеленым – инерциальную силу. В приведенном коде массы показаны единичными векторами, а реакции и сила инерции – с масштабными коэффициентами. Это сделано для наглядности анимации. Все вектора для вывода на экран хранятся в переменных с контекстом *G*`:

```
G`Rb[t_,omega_]={Blue,Arrow[{ob[t,omega],ob[t,omega]+
f`Rb[t,omega]/70}]};
G`Ra[t_,omega_]={Blue,Arrow[{oa[t,omega],oa[t,omega]+
f`Ra[t,omega]/70}]};
G`Ro[t_,omega_]={Blue,Arrow[{{0,0}, f`Ro[t,omega]/70}]};
G`WcI[t_,omega_]={Green,Arrow[{oc[t,omega],oc[t,omega]-
(m2/.cnst3)*wc[t,omega]/5}]};
G`m1[t_,omega_]={Red,Arrow[{oa[t,omega]*0.5,oa[t,omega]*
0.5+mass}]};
G`m2[t_,omega_]={Red,Arrow[{oc[t,omega],oc[t,omega]+mass
}]};
G`m3[t_,omega_]={Red,Arrow[{ob[t,omega],ob[t,omega]+mass
}]};

Manipulate[Graphics[{G`KPM[t,omega], G`Rb[t,omega],
G`Ra[t,omega], G`Ro[t,omega], G`WcI[t,omega],
```

```
G`m1[t,omega], G`m2[t,omega], G`m3[t,omega]},PlotRange-
>{{-1,3},{-2,2}}],{t,0,100,0.125},{omega,1,5,1}]
```

Чтобы проверить корректность расчетов, построим план сил второго звена. Здесь будем использовать ту же цветовую схему, как и ранее. В результате мы должны получить замкнутый многоугольник (см. рис. 18).

```
G`F1[t_,omega_]={Blue,Arrowheads[0.2],Arrow[{{0,0},
f`Ra[t,omega]}]};
G`F2[t_,omega_]={Red,Arrowheads[0.2],Arrow[{Raf[t,omega]
, f`Ra[t,omega]+m2*g*mass}]};
G`F3[t_,omega_]={Red,Arrowheads[0.2],Arrow[{f`Ra[t,omega
]+m2*g*mass, f`Ra[t,omega]+m2*g*mass+m3*g*mass}]};
G`F4[t_,omega_]={Green,Arrowheads[0.2],Arrow[{f`Ra[t,ome
ga]+m2*g*mass+m3*g*mass,
f`Ra[t,omega]+m2*g*mass+m3*g*mass-m2*wc[t,omega]}]};
G`F5[t_,omega_]={Blue,Arrowheads[0.2],Arrow[{f`Ra[t,omeg
a]+m2*g*mass+m3*g*mass-m2*wc[t,omega],
f`Ra[t,omega]+m2*g*mass+m3*g*mass-m2*wc[t,omega]+
f`Rb[t,omega]}]};
```

```
Manipulate[Graphics[{G`F1[t,omega], G`F2[t,omega],
G`F3[t,omega], G`F4[t,omega],
G`F5[t,omega]}],{t,0,100,0.125},{omega,1,5,1}]
```

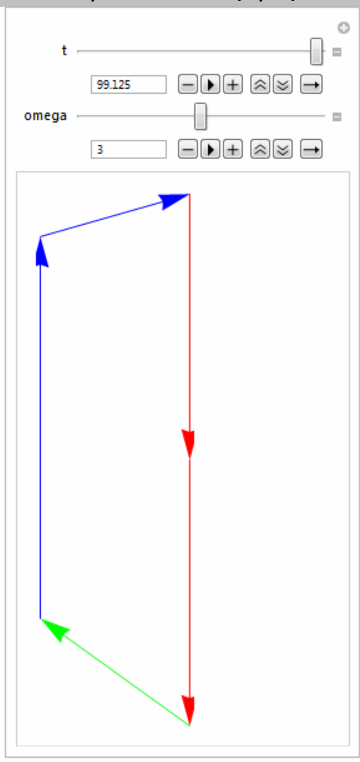


Рис. 18. План сил

В случае увеличения количества учитываемых сил трудоемкость вышеуказанного кода значительно увеличивается, поэтому целесообразней обобщить построение плана сил с помощью массива. Определим список (массив) *PF* для сил:

```
PF={{0,0},f`Ra[t,omega],m2*g*mass/.cnst3,m3*g*mass/.cnst
3,-m2*wc[t,omega]/.cnst3,f`Rb[t,omega]};
```

Создадим список цветов для каждой силы (здесь *Null* означает пустое значение):

```
color={Null,Blue,Red,Red,Green,Blue};
```

Начальная точка каждого вектора определяется как сумма всех предыдущих векторов, тогда используем функцию *Sum[f,{i,*i_{min}*,*i_{max}*,di}],* которая возвращает сумму выражения *f* по *i* в интервале *{*i_{min}*,* i_{max}*}* с шагом *di*. С помощью упомянутой функции определим массив изображений векторов сил *G`F:*

```
G`F[t_,omega_]=Table[{color[[i]],Arrowheads[0.2],Arrow[{
Sum[PF[[j]],{j,1,i-1}],Sum[PF[[j]],{j,1,i}]}]},{i,2,6}];
```

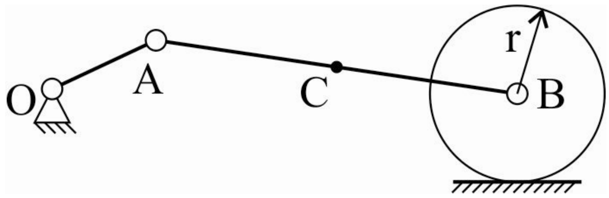
Теперь выведем массив *G`F* на экран:

```
Manipulate[Graphics[G`F[t,omega]],{t,0,100,0.125},{omega
,1,5,1}]
```

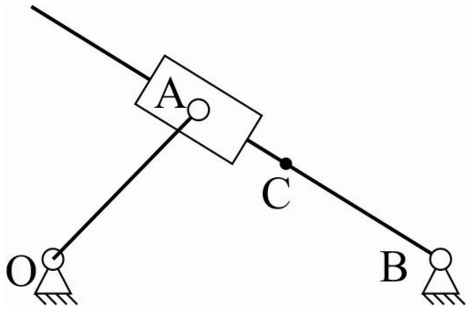
Задание: *постройте план сил всей конструкции.*

Задание: *решите задачу с учетом силы инерции третьего звена – ползуна.*

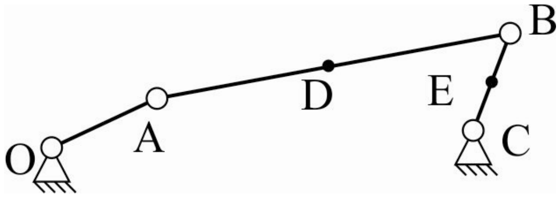
ЗАДАНИЯ



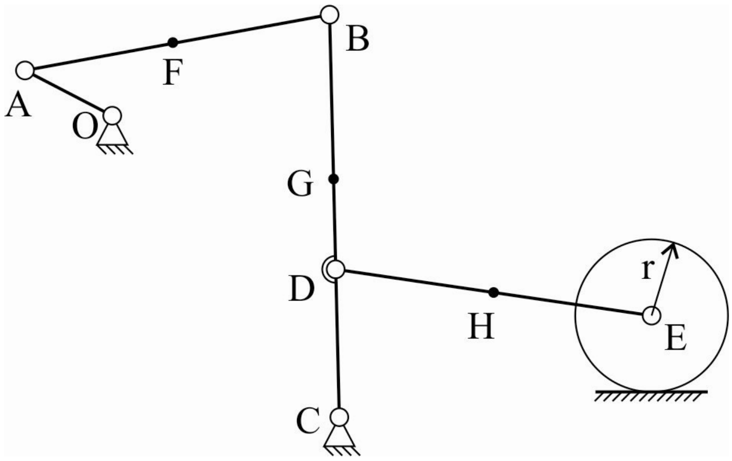
Определите кинематические характеристики механизма и реакции в связях: OA=0.5AB=0.3r, AC=CB. Ведущее звено OA вращается с постоянной угловой скоростью ω.



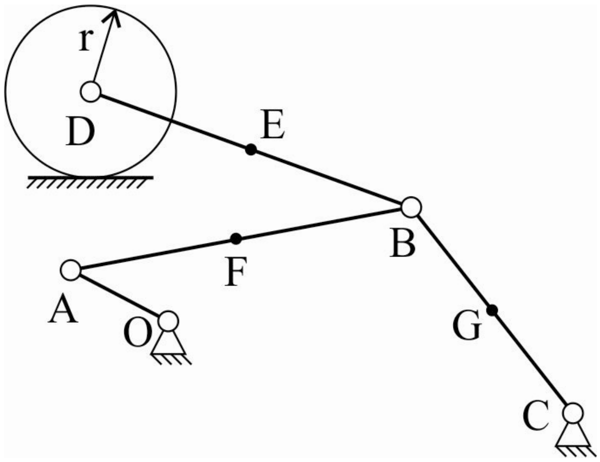
Определите кинематические характеристики механизма и реакции в связях: OA=BC. Ведущее звено OA вращается с постоянной угловой скоростью ω.



Определите кинематические характеристики механизма и реакции в связях: OA==0.3AB=0.5BC, AD=DB, CE=EB. Ведущее звено OA вращается с постоянной угловой скоростью ω.



Определите кинематические характеристики механизма и реакции в связях: OA==0.5AB=0.25BC=0.5DE=0.75r, AF=FB, CG=GB, CD=0.25CB, DH=HE. Ведущее звено OA вращается с постоянной угловой скоростью ω.



Определите кинематические характеристики механизма и реакции в связях: OA==0.5AB=0.5BC=0.5BD=0.75r, AF=FB, CG=GB, BE=DE. Ведущее звено OA вращается с постоянной угловой скоростью ω.

ПРИЛОЖЕНИЕ

1. Работа с файлами

На практике часто возникает задача чтения результата эксперимента из файла для сравнения с полученными результатами моделирования или же задача сохранения полученных данных в файл для использования их в последующих расчетах. Рассмотрим возможности среды Mathematica для работы с файлами. Для определения рабочей папки файлов используется команда *SetDirectory[dir]*, которая устанавливает папку по умолчанию *dir*. Команда *Directory[]* возвращает актуальную рабочую папку.

2. Экспорт

Экспорт данных из Mathematica реализуется командой *Export["file.ext", expr, "format"]* которая записывает данные *expr* в файл *file* с расширением *ext* в формате *format* (этот параметр необязателен для задания). Часто используемые параметры для *format*: *"Data"* – упорядоченные данные, *"Image"* – графический объект,*"Graphics"* и *"Graphics3D"* – график или геометрия в 2D или 3D соответственно (остальные типы форматов смотрите в документации к Mathematica). Рассмотрим экспорт данных на примере:

```
Export["plot.jpg",Plot[Sin[x],{x,0,Pi}]]
Export["plot.jpg",Plot[Sin[x],{x,0,Pi},"Image"]]
```

Здесь в обоих случаях в рабочую папку в файл *plot.jpg* будет записан график функции *Sin(x)* на интервале [0,π]. Mathematica распознает формат по расширению файла, уточнение формата чаще всего необходимо в конкретных случаях.

```
Export["plot.txt",Table[{x,Sin[x]},{x,0,Pi,0.1}]]
```

В этом примере в рабочую папку в файл *plot.txt* будет записана табуляция функции *Sin(x)* на интервале [0,π] с шагом 0.1. Но пары чисел будут заключены в массивные скобки Mathematica. Чтобы убрать их и задать разделителем табуляцию, зададим формат *"Table"*:

```
Ex-
port["plot.txt",Table[{x,Sin[x]},{x,0,Pi,0.1}],"Table"]
```

2. Импорт

Импорт данных в Mathematica реализуется командой *Import["file.ext", elements]*, которая читает данные *elements* (необязательный параметр) из файла *file* с расширением *ext*. Прочтем ранее записанные данные:

```
Import["plot.jpg"]
Import["plot.jpg","ImageSize"]
```

В первом случае результатом будет изображение графика функции, во втором – размер изображения. Теперь перейдем к импорту данных. Следующая команда загрузит текстовые данные табуляции функции из файла и запишет их в переменную *T*:

```
T=Import["plot.txt"]
```

При таком чтении из файла данные в *T* не воспринимаются как массив – для этого необходимо использовать опцию *Table*:

```
T=Import["plot.txt","Table"];
```

Теперь данные из файла прочтены и записаны в *T* в виде массива, значит, к ним применимы все действия, возможные с массивами данных:

```
T//MatrixForm
T[[2]]
T[[2,2]]
```

Второй командой, полезной для чтения из файла, является *ReadList["file.ext",type]*, которая читает данные заданного типа *type* из файла *file* с расширением *ext*. Прочтем те же данные из файла с табуляцией функции и запишем их в массив *T*:

```
T=ReadList["plot.txt",Number,RecordLists->True]
T//MatrixForm
T[[2]]
```

СПРАВОЧНЫЙ МАТЕРИАЛ

context`name	обращение к переменной name по контексту context
name[[i]]	обращение к i-ому элементу массива (списка, вектора) name
func[expr] или *expr//func*	вызов функции *func* с аргументом *expr*
name[[i,j]] или *name[[i]][[j]]]*	обращение к i-ому j-ому элементу массива name
expr/.rule	подстановка в выражение expr правила rule
expr//.rule	повторяющаяся подстановка в выражение expr правила rule
name[arg_]=expr	задание функции name с аргументом arg в виде выражения expr
a.b	скалярное умножение вектора a на b
Cross[a,b]	векторное умножение вектора a на b
Norm[a]	норма объекта a
Solve[expr, vars]	решение заданного уравнения expr относительно списка неизвестных vars
Graphics[obj]	вывод на экран графического объекта
$Plot[f,\{x,x_{min},x_{max}\}]$	построение графика функции f=f(x) в диапазоне (x_{min},x_{max})
$ParametricPlot[f,\{t,t_{min},t_{max}\}]$	построение графика функции $f_i=f_i(t)$ в диапазоне (t_{min},t_{max})
D[f,{x,n}]	n-ая производная от функции f
$Table[expr,\{i,i_{min},i_{max},di\}]$	возвращает список выражения expr(i) в диапазоне (i_{min},i_{max}) с шагом di
$Sum[expr,\{i,i_{min},i_{max},di\}]$	возвращает сумму выражения expr(i) в диапазоне (i_{min},i_{max}) с шагом di

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Дьяконов В.П. Mathematica 5.1/5.2/6 Программирование и математические вычисления. – М.: LVR-Пресс, 2008. – 576 с.
2. Дьяконов В.П. Mathematica 5/6/7 Полное руководство. – М.: LVR-Пресс, 2010. – 624 с.
3. Кристалинский Р.Е., Шапошников Н.Н. Решение варионных задач строительной механики в системе Mathematica: *Учебное пособие*. – Спб: Издательство "Лань", 2010. – 250 с.
4. Артюхин Ю.П. Строительная механика в пакетах "MATHEMATICA" и "ANSYS": *учебное пособие*. Казань, 2009. – 120 с.
5. Артюхин Ю.П., Гурьянов Н.Г., Котляр Л.М. Система Математика 4.0 и ее приложения в механике: *учебное пособие*. Казань, Набережные Челны, 2002. – 415 с.
6. Новожилов И.В., Зацепин М.Ф. Типовые расчеты по теоретической механике на базе ЭВМ: *Учеб. пособие для втузов*. – М.: Высш. шк., 1986 – 136 с.: ил.