

Использование компьютерных моделей при решении задач по теме “Динамика тела”

Майер Р.В.

Аннотация—В статье рассмотрены решения следующих пяти задач по механике с помощью ПЭВМ: 1) качение цилиндра по внутренней поверхности вращающейся трубы; 2) падение упругой линейки на горизонтально расположенный цилиндр; 3) падение длинной упругой нити на горизонтально расположенный цилиндр, а затем на поверхность стола; 4) движение резинового кольца, которое сначала растянули, а затем отпустили; 5) движение вращающегося кольца, падающего на горизонтальную поверхность. Представлены четыре программы, написанные в среде Free Pascal, приведены результаты моделирования.

Ключевые слова—компьютерное моделирование, программирование, методика преподавания физики, решение задач по механике.

Применение численных методов позволяет существенно расширить круг решаемых задач [1–7]. Актуальной является проблема создания банка несложных компьютерных программ, позволяющих промоделировать различные

физические явления и решить соответствующие учебные задачи. Желательно, чтобы программы были написаны в распространенной среде программирования (например, Free Pascal). Применение компьютерных моделей способствует установлению межпредметных связей между математикой, физикой, информатикой; они могут быть использованы при проведении учебных исследований, в курсовых и дипломных работах. Рассмотрим несколько задач по динамике тела.

Задача 1. По внутренней поверхности вращающейся трубы радиуса R катится цилиндр радиуса r , ось A которого соединена стержнем OA с точкой O (рис. 1.1). Скорость вращения трубы изменяется по закону $\omega = \omega(t)$. Между поверхностью трубы и цилиндром, а также в подшипниках O и A действует сила вязкого трения (коэффициенты трения r_s и k). Необходимо рассчитать движение системы.

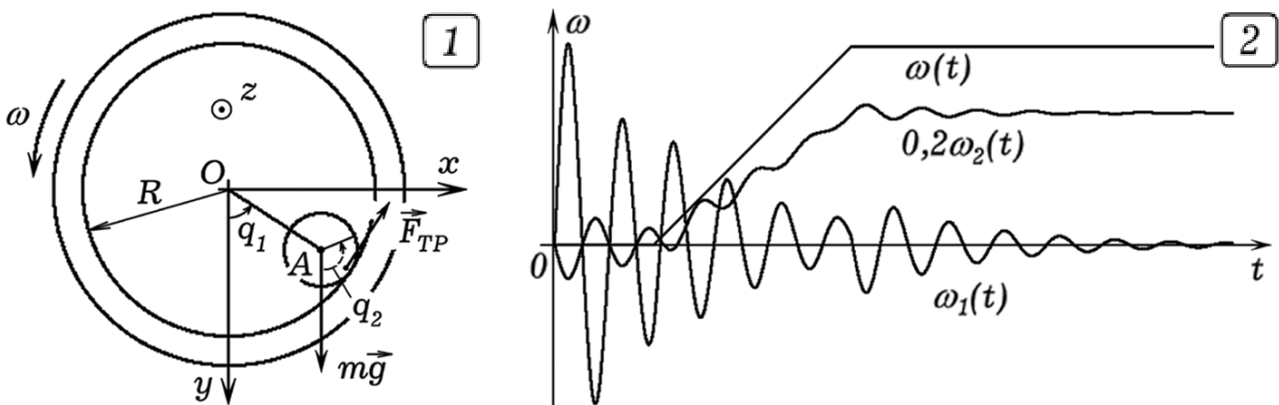


Рис. 1. Расчет качения цилиндра во вращающейся трубе.

Система имеет две степени свободы. Введем две обобщенные координаты q_1 и q_2 (в рад). Запишем систему уравнений Лагранжа:

$$L = \frac{m}{2} \dot{q}_1^2 (R-r)^2 + \frac{I}{2} \dot{q}_2^2 + mg(R-r) \cos q_1,$$

$$m\ddot{q}_1(R-r)^2 + mg(R-r)^2 \sin q_1 = F_{TP}R - k\dot{q}_1,$$

$$I\ddot{q}_2 = F_{TP}r - k\dot{q}_2, \quad F_{TP} = r_s(\omega R - \dot{q}_1 R - \dot{q}_2 r).$$

При этом учитывается, что сила вязкого трения между поверхностями трубы и цилиндра

пропорциональна скорости движения точек одной поверхности относительно другой. Тормозящий момент, действующий в подшипниках пропорционален угловой скорости вращения.

```
{N+}uses crt,graph; {PP-1}
const dt=1E-4; X0=320; Y0=240; R=100;
rc=30; m=0.3; rs=5; k=50; I=0.5;
var Gd,Gm,tt: integer; q1,q2,fi,
w,w1,w2,e1,e2,t,xc,yc,Ftr: single;
```

```

BEGIN Gd:=Detect; InitGraph(Gd, Gm,
'c:\bp\bgi'); q1:=-2;
Repeat t:=t+dt; inc(tt); w:=2*(t-4); If w>10 then
w:=10; If t<4 then w:=0; Ftr:=rs*(w*R-w1*R-w2*rc);
e1:=(Ftr*R-k*w1-m*980*(R-rc)*sin(q1))/m/sqr(R-rc);
e2:=(Ftr*rc-k*w2)/I; w1:=w1+e1*dt; w2:=w2+e2*dt;
q1:=q1+w1*dt; q2:=q2+w2*dt; fi:=fi+w*dt;
{circle(round(t*20),240-round(w*10),1); circle
(round(t*20),240-round(w1*10),1); circle(round
(t*20),240-round(w2*2),1);}
If tt mod 20=0 then begin tt:=0;
cleardevice; circle(X0,Y0,round(R));
xc:=X0+(R-rc)*sin(q1); yc:=Y0+(R-rc)*cos(q1);
line(round(xc),round(yc),X0,Y0); circle(X0+
round(110*sin(fi)),Y0+round(110*cos(fi)),2); circle
(round(xc),round(yc),round(rc)); line (round(xc),
round(yc),round(xc+23*sin(q2)),round(yc+23*
cos(q2))); circle(round(xc+24*sin(q2)),round(yc+24*
cos(q2)),1); end;
until KeyPressed; CloseGraph;
END.

```

Используется программа ПР-1 (среда Free Pascal), результаты моделирования движения цилиндра представлены на рис. 1.2. Допустим, что в начальный момент система находилась в состоянии $q_1 = -1$ рад, $q_2 = 0$ и ω_{1Z} и $\omega_{2Z} = 0$, причем труба неподвижна. Систему предоставляют самой себе, и через некоторое время t' скорость вращения трубы ω начинает расти пропорционально времени до тех пор, пока не достигнет заданного значения ω' , после чего остается постоянной. Из графиков, изображенных на рис. 1.2, понятно, как изменяются проекции угловой скорости движения кривошипа OA ω_{1Z} и скорость вращения цилиндра ω_{2Z} с течением времени. Так как система выведена из положения равновесия, то кривошип (а значит и ω_{1Z}) сначала совершает затухающие колебания; после разгона трубы он продолжает колебаться относительно положения, смещенного от положения равновесия. Угловая скорость цилиндра сначала также совершает затухающие колебания, а после разгона трубы увеличивается, стремясь к некоторому положительному значению. При этом цилиндр смещен из нижнего положения вправо и вращается против часовой стрелки (в этом можно убедиться, построив график $q_1(t)$).

Задача 2. Упругая линейка (стержень) падает на горизонтально расположенный цилиндр, отскакивает от него и затем падает на горизонтальную поверхность. Необходимо рассчитать положение линейки в последовательные моменты времени. Движение происходит в вертикальной плоскости.

Будем использовать **метод больших частиц**. Вместо линейки рассмотрим совокупность твердых макроскопических шариков или дисков, взаимодействующих друг с другом с заданными

силами упругости \vec{F}_{ij} и внутреннего трения \vec{F}_{ij}' . Расстояние между центрами частиц обозначим через r . Проекцию F_r силы взаимодействия на ось r зададим так: 1) если $r > 10$, то частицы не взаимодействуют: $F_r = 0$; 2) если r лежит в интервале $[6; 10]$, то действует сила упругости $F_r = 1000 \cdot (8 - r)$; 3) если $r < 6$, то частицы сильно отталкиваются: $F_r = 10000$. Величины r и F_r заданы в условных единицах. Сила внутреннего трения \vec{F}_{ij}' пропорциональна скорости смещения i -той частицы относительно j -той и направлена противоположно. Проекция силы с которой j -тая частица действует на i -тую на ось Ox (рис. 2.1):

$$F_{ijx} = (F_{ij} - r_c(l_{ij}^{t+1} - l_{ij}^t) / \Delta t) \cos \alpha_{ij},$$

$$\cos \alpha_{ij} = (x_j - x_i) / l_{ij}.$$

Здесь l_{ij}^t и l_{ij}^{t+1} — расстояния между частицами в моменты t и $t+1$. Проекция F_{ijy} рассчитывается аналогично.

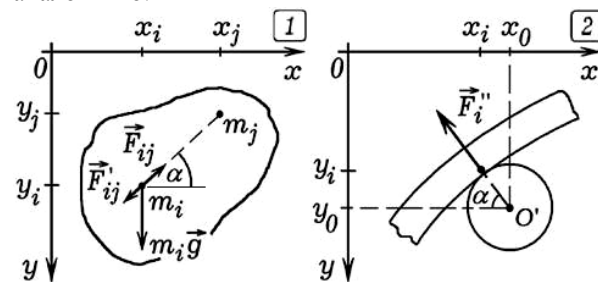


Рис. 2. К решению задачи о линейке (стержне), падающей на цилиндр.

При соударении линейки с горизонтально расположенным цилиндром радиусом R на i -тую частицу, соприкасающуюся с цилиндром, действует сила F_i'' (рис. 2.2). В программе ПР-2 рассчитываются синус и косинус угла α между прямой, идущей от центра цилиндра к этой i -той частице, и горизонтальной осью Ox :

$$\cos \alpha = (x_0 - x_i) / R, \quad \sin \alpha = (y_0 - y_i) / R.$$

Если i -тая частица приблизилась к точке O' на расстояние меньшее R , то программа "отодвигает" частицу на расстояние чуть большее R и уменьшает ее скорость.

```

{$N+} uses crt, graph; {ПР-2}
const N=80; r=1E+5; dt=0.0003; X0=490; Y0=200;
var Fx,Fy,x,y,vx,vy: array[1..N] of single;
s1,s: array[0..N,0..N] of single; time: string;
t: Longint; Gd,Gm,i,j,d,rad: integer;
m,rr,tt,FFx,FFy,FF,k,kk,ax,ay,F,l: single;
Procedure Sila; label Metka;
begin For i:=1 to N do begin Fx[i]:=0; Fy[i]:=0; end;
For i:=1 to N do For j:=1 to N do begin
If (j=i) or (s[i,j]>16) then goto Metka;
l:=sqrt(sqrt(x[i]-x[j])+sqrt(y[i]-y[j]));
F:=20000*(s[i,j]-l);

```

```

If l<6 then F:=10000; FFx:=0; FFy:=0; m:=1;
Fx[i]:=Fx[i]+(F-r*(1-s1[i,j]))*(x[i]-x[j])/(1+1E-3)+FFx;
Fy[i]:=Fy[i]+(F-r*(1-s1[i,j]))*(y[i]-y[j])/(1+1E-3)+
m*2.3+FFy; Metka: end; end;
Procedure Nach_uslov; begin Randomize; d:=2;
For j:=1 to 40 do begin y[j]:=80; x[j]:=8*j+320;
y[40+j]:=88; x[40+j]:=8*j+320; end;
For i:=1 to N do For j:=1 to N do begin
s[i,j]:=sqrt(sqrt(x[i]-x[j])+sqrt(y[i]-y[j])); end; end;
BEGIN Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
Nach_uslov; Randomize;
Repeat Sila; tt:=tt+dt;
For i:=1 to N do For j:=1 to N do begin
s1[i,j]:=sqrt(sqrt(x[i]-x[j])+sqrt(y[i]-y[j])); end;
For i:=1 to N do begin ax:=Fx[i]/m; ay:=Fy[i]/m;
vx[i]:=vx[i]+ax*dt; vy[i]:=vy[i]+ay*dt;
x[i]:=x[i]+vx[i]*dt; y[i]:=y[i]+vy[i]*dt;
If y[i]>410 then begin y[i]:=409;
vy[i]:=-0.2*vy[i]; vx[i]:=-0.2*vx[i]; end;
rr:=sqrt(sqrt(y[i]-Y0)+sqrt(x[i]-X0)); rad:=40;
If rr<rad then begin x[i]:=X0+(rad+0.2)*(x[i]-X0)/rr;
y[i]:=Y0-(rad+0.2)*(Y0-y[i])/rr; vx[i]:=-0.3*vx[i];
vy[i]:=-0.3*vy[i]; end;
end; circle(round(x[N]),round(y[N]),1);
If t mod 1000=0 then begin
t:=0; cleardevice; line(0,412,940,412);
circle(X0,Y0,rad); circle(X0,Y0,rad-2);
STR(round(100*t),time); OutTextXY(130,100,time);
For i:=1 to N do circle(round(x[i]),round(y[i]),3);
end; inc(t);
until KeyPressed; CloseGraph;
END.

```

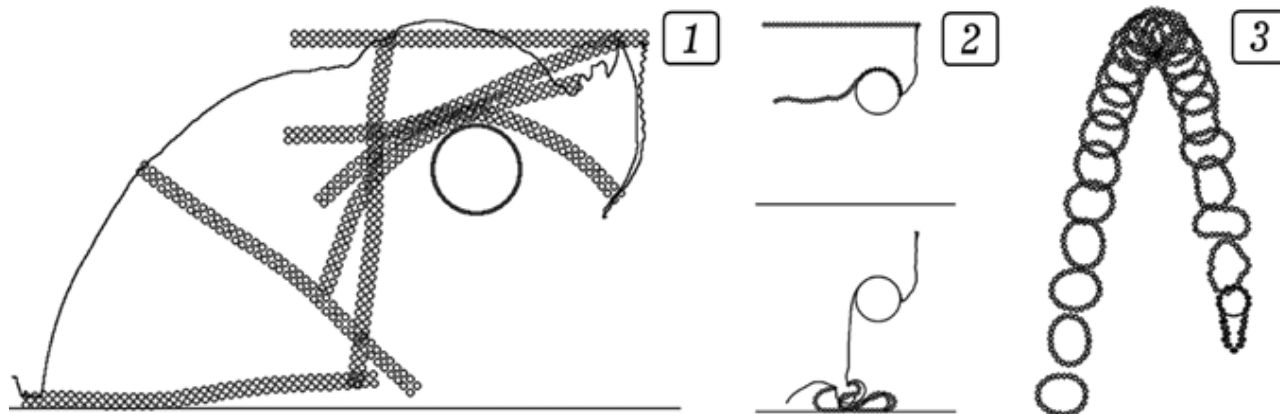


Рис. 3. Моделирование методом больших частиц: 1) падение линейки; 2) падение нити на цилиндр; 3) движение резинового кольца.

Задача 4. Резиновое кольцо надевают на цилиндр (палец) и нижнюю часть растягивают силой, направленной вниз. Затем в момент t_1 кольцо отпускают, а в момент $t_2 > t_1$ убирают цилиндр. Кольцо взлетает вверх. Необходимо промоделировать движение кольца, рассчитать изменение координат и скоростей ее точек с течением времени.

Задача решается тем же способом, нужно только изменить начальное расположение частиц и параметры системы. Чтобы промоделировать

Результаты моделирования движения линейки при падении на цилиндр, а затем на горизонтальную плоскость представлены на рис. 3.1 [3]. На них показаны ее положения в различные моменты времени и траектория одного из концов. Сначала линейка движется практически поступательно, после удара о цилиндр она подпрыгивает и совершает свободные колебания, изгибаясь в ту и другую стороны. Под действием силы тяжести центр масс линейки стремится занять самое низкое положение. Линейка переворачивается, соударяется с горизонтальной плоскостью, по ней бежит изгибающая волна. Через некоторое время за счет действия сил трения система переходит в состояние покоя.

Задача 3. Длинная упругая нить падает на горизонтально расположенный цилиндр, а затем соскальзывает на горизонтальную поверхность. Необходимо рассчитать координаты и скорости различных точек нити в последовательные моменты времени.

В предыдущей задаче линейка моделировалась двумя рядами частиц, между которыми действовали вязкоупругие силы. Чтобы промоделировать нить достаточно рассмотреть один ряд частиц. Во всем остальном задача решается совершенно аналогично. На рис. 3.2 показано положение нити в различные моменты времени, а также траектория движения частицы с номером $i = N$. Видно, как нить охватывает цилиндр, затем соскальзывает с него на горизонтальную поверхность.

движение эластичного кольца следует уменьшить коэффициенты упругости и вязкого трения. Используется программа ПР-3; результаты расчета движения кольца представлены на рис. 3.3. Видно, что после отрыва от цилиндра центр масс кольца движется по параболе, а само кольцо периодически деформируется, совершая затухающие колебания на собственной частоте.

```

{$N+} uses crt, graph;
const N=24; r=2E+3; dt=0.0006;
var Fx,Fy,x,y,vx,vy: array[1..N]of

```

{ПР-3}

```

single; s1,s:array[0..N,0..N]of
single; Gd,Gm,i,j,t,rad: integer;
m,rr,tt,FFx,FFy,FF,k,kk,ax,ay,F,l: single;
Procedure Sila; label Metka;
begin For i:=1 to N do begin
Fx[i]:=0; Fy[i]:=0; end;
For i:=1 to N do For j:=1 to N do
begin If (j=i)or(s[i,j]>25) then goto Metka;
l:=sqrt(sqr(x[i]-x[j])+sqr(y
[i] - y[j])); F:=2000*(s[i,j]-l);
If l<5 then F:=10000; FFx:=0; FFy:=0;
If (i=5)and(tt<5) then FFy:=5000 else FFy:=0;
Fx[i]:=Fx[i]+(F-r*(l-s1[i,j]))*(x[i] -x[j])/(l+1E-3)+FFx;
Fy[i]:=Fy[i]+(F-r*(l-s1[i,j]))*(y[i]-y[j])/(l+1E-3)
+m*0.2+FFy; Metka: end; end;
Procedure Nach_uslov;
begin Randomize;
For j:=1 to 24 do begin y[j]:=400+30* sin(6.28*j/24);
x[j]:=320+30*cos(6.28*j/24); end;
For i:=1 to N do For j:=1 to N do s[i,j]:=
sqrt(sqr(x[i]-x[j])+sqr(y[i] -y[j])); end;
BEGIN Gd:=Detect; InitGraph(Gd,Gm,'c:\bp\bgi');
Nach_uslov; Randomize;
Repeat Sila; tt:=tt+dt; inc(t);
For i:=1 to N do For j:=1 to N do s1[i,j]:=sqrt(sqr(x[i]-
x[j])+sqr( y[i]-y[j])); For i:=1 to N do begin
If i=8 then m:=5 else m:=1; ax:=Fx[i]/m; ay:=Fy[i]/m;
vx[i]:=vx[i]+ax*dt; vy[i]:=vy[i]+ay*dt;
x[i]:=x[i]+vx[i]*dt; y[i]:=y[i]+vy[i]*dt;
If tt<5.2 then begin rr:=sqrt(sqr(y[i]-400)+

```

```

sqr(x[i]-320)); rad:=20;
If rr<rad then begin vx[i]:=0.3*vx[i]; vy[i]:=0.3*vy[i];
x[i]:=320+(rad+0.2)*(x[i]-320)/rr;
y[i]:=400-(rad+0.2)*(400-y[i])/rr; end; end; end;
If t mod 30=0 then begin t:=0;
cleardevice; circle(320,400,rad); line(round(x[1]),
round(y[1]),round(x[13]),round(y[13]));
For i:=1 to N do circle(round(x[i]),round(y[i]),3); end;
until KeyPressed; CloseGraph;
END.

```

Задача 5. Частично упругое кольцо бросают на горизонтальную поверхность в поле тяжести, сообщив ему некоторую угловую скорость. Необходимо промоделировать трехмерное движение кольца, рассчитать координаты его точек и их скорость в последующие моменты времени.

Для решения этой задачи используется тот же метод: кольцо заменяется совокупностью 48 (или 24) точек, расположенных по окружности и соединенных между собой вязкоупругими связями. Используется программа ПР-4, результаты моделирования представлены на рис. 4. Если упростить программу, исключив движение вдоль оси Oz , направленной перпендикулярно к рисунку, то удастся промоделировать двумерное движение кольца в вертикальной плоскости при его падении на горизонтальную поверхность.

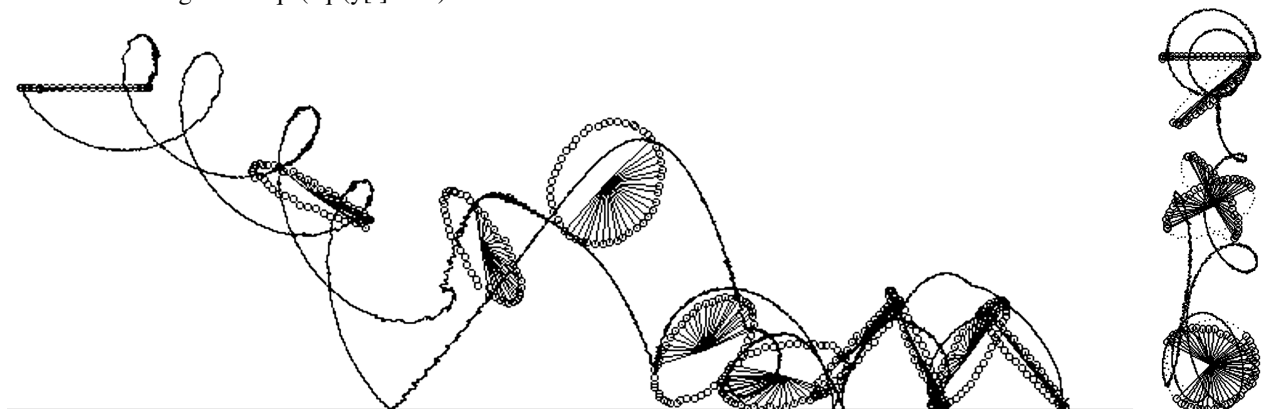


Рис. 4. Падение кувыркающегося кольца на горизонтальную поверхность.

```

{$N+} uses crt, graph; {ПР-4}
const N=48; r=4.5E+5; dt=4E-4; pi=3.1416;
var Fx,Fy,Fz,x,y,z,vx,vy,vz: array [1..N]of single;
s1,s:array[0..N,0..N]of single; Gd,Gm,i,j,t: integer;
z0,m,l,tt,r1,ax,ay,az,F: single;
Procedure Sila; Label MM;
begin For i:=1 to N do begin
Fx[i]:=0; Fy[i]:=0; Fz[i]:=0; end;
For i:=1 to N do For j:=1 to N do
begin If j=i then goto MM;
l:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j])+sqr(z[i]-z[j]));
F:=5E+4*(s[i,j]-l); If s[i,j]<25 then F:=2E+6*(s[i,j]-l);
If l<3 then F:=5E+4; If i=18 then m:=3 else m:=1;
If y[i]>399 then r1:=0.6 else r1:=0.005;
Fx[i]:=Fx[i]+(F-r*(l-s1[i,j]))*(x[i]-x[j])/(l+1E-3)-
r1*vx[i]; Fz[i]:=Fz[i]+(F-r*(l-s1[i,j]))*(z[i]-z[j])/(l+1E
-3)-r1*vx[i]; Fy[i]:=Fy[i]+(F-r*(l-s1[i,j]))*(y[i]-y[j])

```

```

/(l+1E-3)-r1*vx[i]+m*2; MM: end; end;
Procedure Nach_uslov;
begin Randomize;
For j:=1 to N do begin z[j]:= 40*sin(2*pi*j/48);
y[j]:=200; x[j]:=60+40*cos(2*pi*j/48); vx[j]:=100;
end; vy[18]:=3000; vy[48]:=-vy[18];
For i:=1 to N do For j:=1 to N do
s[i,j]:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j])+sqr(z[i]-z[j]));
end;
BEGIN Gd:=Detect; InitGraph(Gd,Gm,""); Nach_uslov;
Repeat tt:=tt+dt; z0:=0;
For i:=1 to N do For j:=1 to N do
s1[i,j]:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j])+sqr(z[i]-z[j]));
Sila; For i:=1 to N do begin ax:=Fx[i]/m;
ay:=Fy[i]/m; az:=Fz[i]/m; vx[i]:=vx[i]+ax*dt;
vy[i]:=vy[i]+ay*dt; vz[i]:=vz[i]+az*dt;
x[i]:=x[i]+vx[i]*dt; y[i]:=y[i]+vy[i]*dt;

```

```

z[i]:=z[i]+vz[i]*dt; z0:=z0+z[i]/N;
If y[i]>400 then begin y[i]:=400; vy[i]:=-0.2*vy[i];
vx[i]:=0.6*vz[i]; vz[i]:=0.6*vz[i]; end; end;
circle(round(x[1]),round(y[1]),1);
circle(round(x[24]),round(y[24]),1);
If t mod 40=0 then begin t:=0;
cleardevice; line(0,400,800,400);
For j:=1 to 24 do line(round((x[1]+x[25])/2),
round((y[1]+y[25])/2),round(x[j]),round(y[j]));
For i:=1 to N do begin
circle(round(x[i]),round(y[i]),2); end; end; inc(t);
until KeyPressed; CloseGraph;
END.

```

Хотя рассмотренные методы моделирования известны [2, 5–7], учебник, в котором были бы проанализированы примеры их использования для решения учебных задач по механике, в настоящее время отсутствует. Проблема использования компьютерных моделей при изучении механики давно обсуждается на различных научно-методических конференциях (например, доклады Саранина В.А. и Данилова О.Е. на конференции “Учебный физический эксперимент: Актуальные проблемы. Современные решения.”, Глазовский пединститут), однако докладчики часто ограничиваются рассмотрением общих идей моделирования и получающихся результатов, но не спешат опубликовать используемую компьютерную программу. В настоящей статье представлены четыре не сложные программы на языке Pascal, которые написаны автором для обучения основам компьютерного моделирования.

Рассмотренные выше компьютерные модели способствуют повышению интереса к механике и

информационным технологиям. Определенный интерес представляет собой проблема создания учебника по компьютерному моделированию для педагогических вузов. Автором статьи уже написаны 15 глав, в которых рассмотрены непрерывные и дискретные, детерминированные и стохастические модели; движение системы частиц в силовом поле; колебательное и волновое движение; системы, состоящие из большого числа частиц; явления переноса, автоволновые процессы; расчет течения жидкости, конвекция; расчет электрического и магнитного полей, движение заряженных частиц (<http://maier-rv.glazov.net>).

БИБЛИОГРАФИЯ

1. Гулд Х., Тобочник Я. Компьютерное моделирование в физике: В 2-х частях. Часть 1. — М.: Мир, 1990. — 350 с.
2. Кунин С. Вычислительная физика. — М.: Мир, 1992. — 518 с.
3. Майер Р. В. Задачи, алгоритмы, программы / Р. В. Майер [Электронный ресурс]. URL: <http://maier-rv.glazov.net>, <http://mayer.hop.ru>
4. Майер Р.В. Компьютерное моделирование физических явлений. — Глазов, ГГПИ: 2009. — 112 с. (<http://maier-rv.glazov.net>)
5. Поттер Д. Вычислительные методы в физике.— М.: Мир, 1975.— 392 с.
6. Giordano N.J. Computational Physics. — New Jersey, Prentice Hall, 1997. — 419 p.
7. Woolfson M.M., Pert G.J. An Introduction to Computer Simulation. — Oxford University Press, 1999. — 311 p.

Use of computer models at the solution of tasks on a subject "Dynamics of a body"

Mayer R.V.

Abstract-In article solutions of the following five tasks on mechanics by means of the computer are considered: 1) rolling of cylinder on an internal surface of a rotating pipe; 2) falling of an elastic ruler on horizontally located cylinder; 3) falling of a long elastic thread on horizontally located cylinder, and then on a table surface; 4) movement of a rubber ring which at first dragged out, and then released; 5) movement of the rotating ring falling on a horizontal surface. Four programs written in the Free Pascal are submitted, results of modeling are given.

Keywords-computer modeling, programming, a technique of teaching of physics, the solution of tasks on mechanics