

Майер Р.В.
КОМПЬЮТЕРНЫЕ МОДЕЛИ ФИЗИЧЕСКИХ ЯВЛЕНИЙ
 Глазовский государственный педагогический институт

Компьютерное моделирование — один из важнейших и интереснейших вопросов курса информатики. В статье рассмотрены компьютерные модели некоторых физических явлений, которые могут быть использованы на уроках информатики, физики и математики, а также на занятиях в вузе. В их основе лежит метод конечных разностей Эйлера. Все программы написаны на языке Pascal.

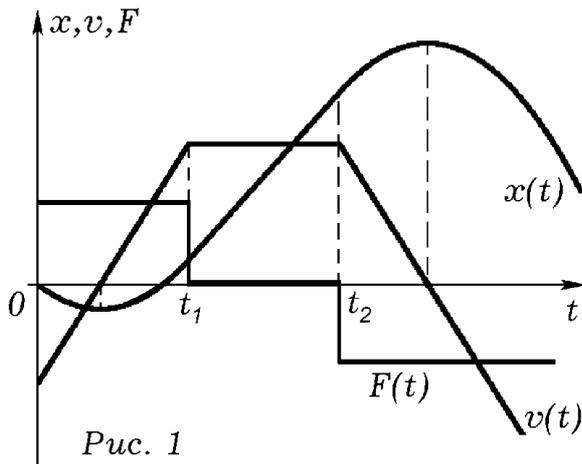
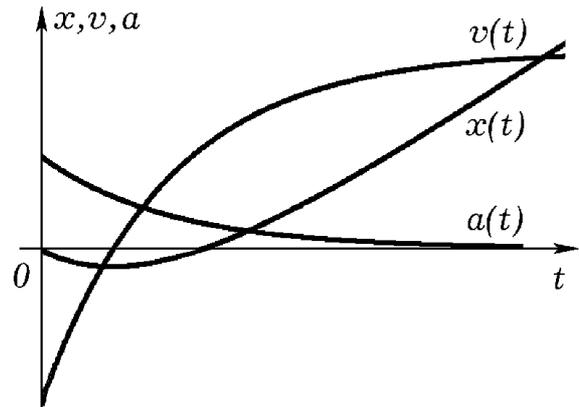


Рис. 1



Задача 1. Система с одной степенью свободы. Имеется физическая система с одной степенью свободы, состоящая из материальной точки известной массы, подвешенной на пружине в вязкой среде. Заданы начальные координата, скорость и действующая сила. Необходимо определить координату и скорость точки в последующие моменты времени.

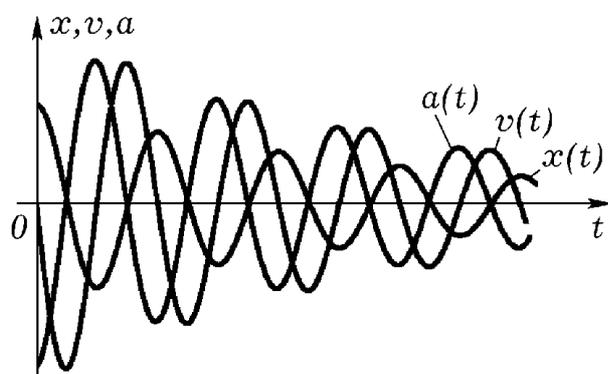
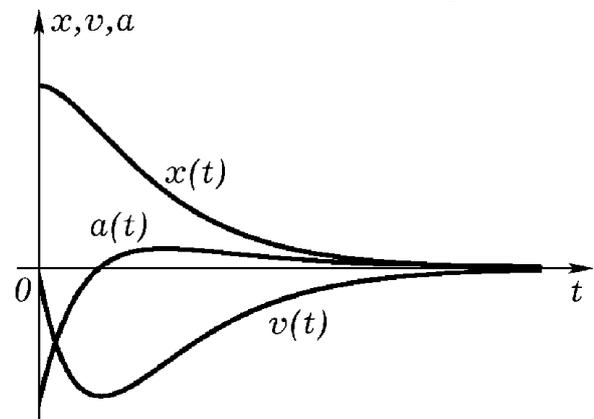


Рис.2



Программа, моделирующая движение рассматриваемой системы представлена ниже. Изменяя параметры системы, закон изменения силы и начальные условия, можно исследовать большую совокупность механических, электрических и других явлений. К ним относятся движение тела под действием изменяющейся силы, падение камня в вязкой среде (рис.1), затухающие колебания (рис.2), а также целый ряд явлений, происходящих в электрических цепях, содержащих резистор, конденсатор и катушку индуктивности (рис.3).

```

program PROGRAMMA1;
uses dos, crt, graph; const Fm=10;w=5;m=2;r=0;k=0; Mx=20; Mv=40; Ma=8; Mf=2; Mt=100;
dt=0.00006; Var x,v,a,F,t : Real; j,xx,vv,aa,FF,tt,Gd,Gm : Integer;
BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi'); if GraphResult<>grOk then Halt(1);
t:=0; v:=0; x:=-3; line(30,300,650,300); line(31,500,31,10); OutTextXY(50,20,'X, V, A');
Repeat {Задание функции F=F(t)}

```

```

begin t:=t+dt; (* F:=Fm*sin(w*t); *) (*If sin(w*t)<0 then F:=0; If sin(w*t)>0 then F:=Fm;*)
  F:=0; If t<1 then F:= Fm; If t>3 then F:=-Fm;
  a:=(F-r*v-k*x)/m; x:=x+v*dt; v:=v+a*dt; tt:=round(t*Mt);
  xx:=round(x*Mx); vv:=round(v*Mv); aa:=round(a*Ma); FF:=round(F*Mf);
  circle(30+tt,300-xx,1); circle(30+tt,300-vv,1); circle(30+tt,300-aa,2);
end; until KeyPressed; CloseGraph;
END.

```

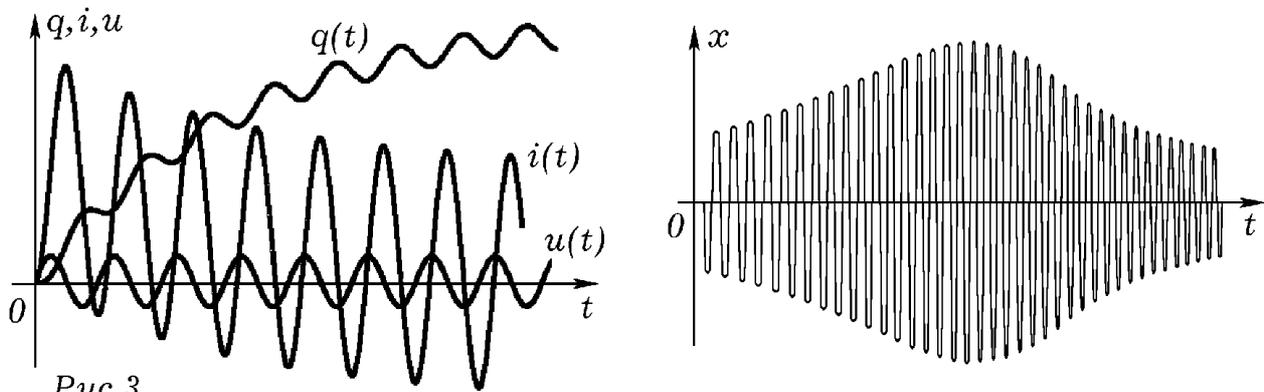


Рис.3

Задача 2. Двумерное движение материальной точки. Материальная точка известной массы движется в заданном силовом поле. На нее действует сила вязкого трения, пропорциональная скорости и направленная противоположно. Необходимо, зная начальные координаты и скорость, построить траекторию движения точки.

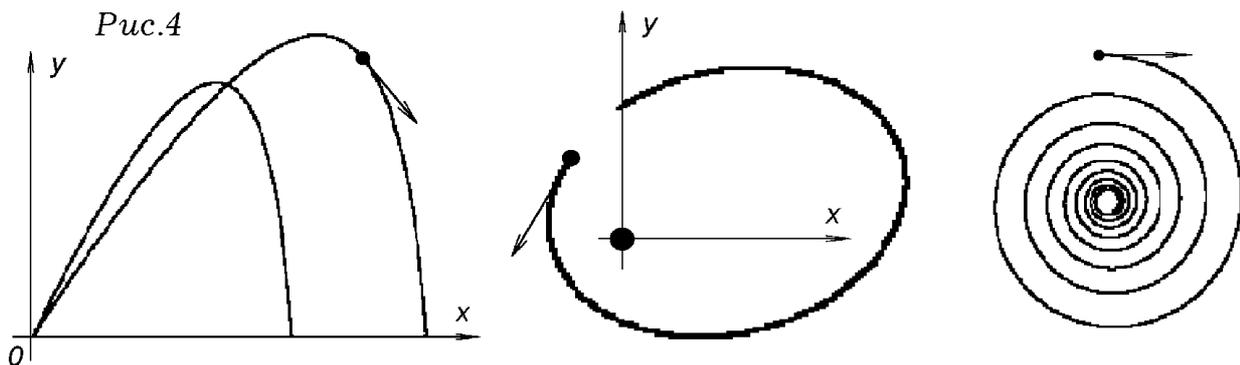


Рис.4

Программа, моделирующая двумерное движение материальной точки в силовом поле, содержит цикл по времени t , в котором вычисляются проекции действующих сил, ускорения и скорости, определяются координаты точки. Модель позволяет изучить движение тела, брошенного под углом к горизонту с учетом вязкости воздуха, движение частицы в поле центральных сил притяжения или отталкивания, промоделировать вращение планет, движение протона в магнитном поле (рис.4) и т.д.

```

program PROGRAMMA2;
uses crt, graph; var v, B, q, F, Fx, Fy : real; r, x, y, vx,vy,ax,ay : real; Gd, Gm, i: integer;
const M=500; mm=100; dt=0.005; rr=0.1; k=2;
BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi'); if GraphResult <> grOk then Halt(1);
  line(320,240,640,240); line(320,240,320,0); circle(320,240,5); x:=100; y:=120; vx:=1; vy:=-2;
Repeat
  begin {--Задание силового поля--}
  (* Fy:=3; Fx:=0;*) (* Fx:=-k*x; Fy:=-k*y; *)
  (* r:=sqrt(x*x+y*y); F:=M*mm/(r*r); Fx:=-F*x/r; Fy:=-F*y/r; *)
  B:=2; q:=1; F:=B*v*q; v:=sqrt(vx*vx+vy*vy); Fx:=F*vy/v; Fy:=-F*v/v;
  (* B:=2; q:=1; F:=B*v*q; v:=sqrt(vx*vx+vy*vy); Fx:=F*vy/v; Fy:=-0.5-F*v/v; *)
  ax:=(Fx-rr*vx)/mm; ay:=(Fy-rr*vy)/mm; vx:=vx+ax*dt; vy:=vy+ay*dt;

```

```

x:=x+vx*dt; y:=y+vy*dt; {--Расчет скоростей и ускорений--}
circle(round(x)+320,240-round(y),2); setcolor(12); circle(round(x)+320,240-round(y),1);
setcolor(15);
end; until KeyPressed; CloseGraph;
END.

```

Задача 3. Движение системы частиц. Имеется система из материальных точек с известными массами, которые взаимодействуют друг с другом и с внешним полем. Зная начальное состояние системы, определите координаты и скорости материальных точек в последующие моменты времени.

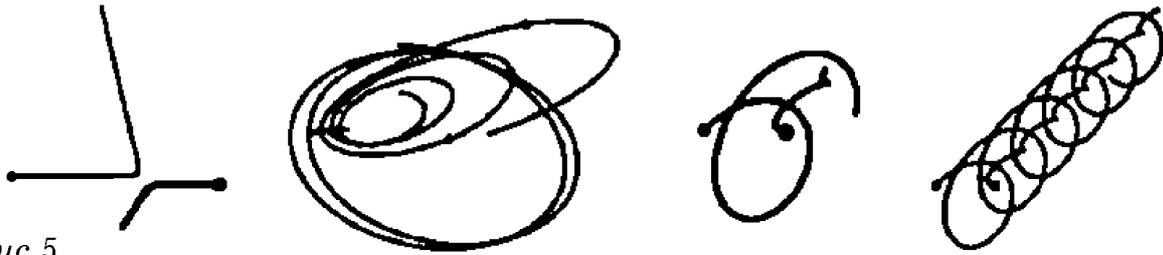


Рис.5

Программа содержит цикл по времени t , в котором определяются равнодействующие всех сил, действующих на каждую материальную точку, вычисляются проекции ускорений и скоростей, находятся координаты материальных точек в следующий момент времени. После этого стираются предыдущие положения частиц и строятся новые. Предлагаемая компьютерная модель позволяет изучить центральный и нецентральный удар, движение двух, трех и более частиц, между которыми действуют силы притяжения или отталкивания (рис.5), движение молекул в сосуде, находящемся в гравитационном поле, диффузию и т.д.

```

program ПРОГРАММА3;
uses dos, crt, graph; const N=50; dt=0.01;
var m,Fx,Fy,x,y,vx,vy,xx,yy : array[1..N] of real;
Gd, Gm, i, j : integer; ax, ay, F, l : real; label Metka, metka1;
Procedure Init_Graph; {Инициализация графики}
begin Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi'); if GraphResult <> grOk then Halt(1); end;
Procedure Sila; {Вычисление действующих сил}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 then goto Metka;
l:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j])); if l<2 then l:=2;
F:=-50000*m[i]*m[j]/sqr(l)+500000*m[i]*m[j]/sqr(l*1);
Fx[i]:=Fx[i]+F*(x[i]-x[j])/l; Fy[i]:=Fy[i]+F*(y[i]-y[j])/l+m[i]*10;
Metka: end; end;
Procedure Nach_uslov;
begin Randomize; {Задание случайных координат и скоростей}
for i:=1 to N do begin m[i]:=2; x[i]:=random(280)+60; y[i]:=random(280)+60;
vy[i]:=random(30)-15; vx[i]:=random(30)-15; end; end;
BEGIN Init_Graph; Nach_uslov; Repeat Sila;
for i:=1 to N do
begin xx[i]:=x[i]; yy[i]:=y[i]; {Запись предыдущих координат}
ax:=Fx[i]/m[i]; ay:=Fy[i]/m[i]; {Вычисление ускорений,}
vx[i]:=vx[i]+ax*dt; vy[i]:=vy[i]+ay*dt; {скоростей,}
x[i]:=x[i]+vx[i]*dt; y[i]:=y[i]+vy[i]*dt; {координат}
if (x[i]<50)or(x[i]>350) then vx[i]:=-vx[i]; {отражение}
if (y[i]<50)or(y[i]>350) then vy[i]:=-vy[i]; {от стенок}

```

```

end; delay(500);
setcolor(8); for i:=1 to N do circle(round(xx[i]),round(yy[i]),2);
setcolor(15); for i:=1 to N do circle(round(x[i]),round(y[i]),2);
until KeyPressed; Repeat until keypressed; CloseGraph;
END.

```

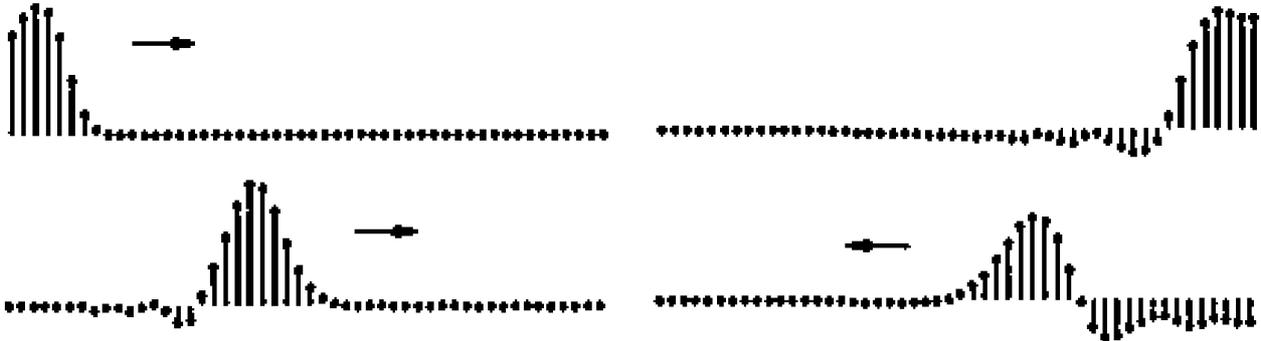


Рис.6

Задача 4. Распространение волн в одномерной упругой среде. Имеется система из осцилляторов с известными массами и жесткостями пружин, которые расположены вдоль прямой и связаны между собой упругими связями. На каждый осциллятор действует сила вязкого трения. Заданы начальные смещения и скорости всех осцилляторов. На отдельные осцилляторы действует вынуждающая сила; некоторые осцилляторы колеблются по заданному закону. Известно, что крайние осцилляторы закреплены (свободны). Необходимо рассчитать координаты и скорости осцилляторов в последующие моменты времени.

Совокупность связанных осцилляторов является системой материальных точек, связанных друг с другом упругими связями, на каждую из которых действует сила вязкого трения. Если один из осцилляторов вывести из состояния покоя, то вдоль всей цепочки побежит волна. Программа, моделирующая это явление, представлена ниже. На рис.6 представлен результат вычислительного эксперимента, в котором изучается отражение импульса от незакрепленного конца струны.

```

program ПРОГРАММА4;
uses dos, crt, graph; const m=0.5; r=0.1; k=0.01; dt=0.001; q=100; N=50;
Var  teta, F, t, y : real; i, xx, vv, aa, FF, tt, Gd, Gm : integer; eta, ksi : array [0..N] of real;
Procedure Graph_Init;
Begin  Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi'); if GraphResult <> grOk then Halt(1); end;
Procedure Oscillator;
Begin  F:=q*(ksi[i-1]-ksi[i])+q*(ksi[i+1]-ksi[i]); teta:=(F-r*eta[i]-k*ksi[i])/m;
eta[i]:=eta[i]+teta*dt; ksi[i]:=ksi[i]+eta[i]*dt; end;
BEGIN  Graph_Init;
Repeat begin t:=t+dt;
For i:=1 to N do begin  y:=ksi[i]; Oscillator;
if 5*t<3.142 then ksi[1]:=20*sin(5*t) else ksi[1]:=0;
ksi[N]:=0; {закреплен} {ksi[N]:=ksi[N-1]; незакреплен}
setcolor(8); circle(10*i, 240-round(y*10),2); setcolor(15); circle(10*i, 240-round(ksi[i]*10),2);
end; end; until KeyPressed; CloseGraph;
END.

```

Задача 5. Двумерная активная среда. Имеется двумерная активная среда, содержащая элементы, каждый из которых может находиться в состояниях покоя, возбуждения и рефрактерности. При отсутствии внешнего воздействия, элемент находится в состоянии покоя. В результате воздействия элемент переходит в возбужденное состояние, приобретая способность возбуждать соседние элементы. Через некоторое время после возбуждения элемент переключается в состояние рефрактерности, находясь в котором он не

может быть возбужден. Затем элемент сам возвращается в исходное состояние покоя, то есть снова приобретает способность переходить в возбужденное состояние. Необходимо промоделировать процессы, происходящие в такой среде при различных ее параметрах и начальном распределении возбужденных элементов.

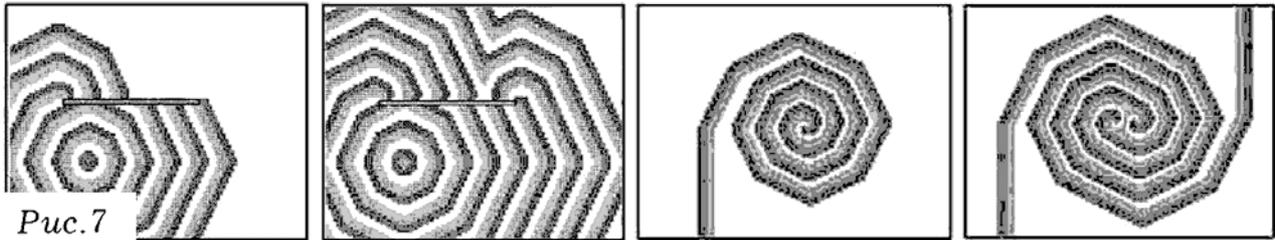


Рис.7

Рассмотрим обобщенную модель Винера—Розенблюта. Мысленно разобьем экран ПК на элементы с индексами i, j , и образующими двумерную сеть. Каждый элемент будем рассматривать как клеточный автомат, состояние которого описывается фазой $y(i, j)$, и концентрацией активатора $u(i, j)$. Если элемент находится в покое, то будем считать, что $y(i, j)=0$. Если вследствие близости возбужденных элементов концентрация активатора $u(i, j)$ достигает порогового значения h , то элемент возбуждается и переходит в состояние 1. Затем на следующем шаге он переключается в состояние 2, затем — в состояние 3 и т.д., оставаясь при этом возбужденным. Достигнув состояния r , элемент переходит в состояние рефрактерности, то есть теряет способность возбуждаться. Через s ($s>r$) шагов после возбуждения элемент возвращается в состояние покоя.

Будем считать, что при переходе из состояния s в состояние покоя 0 концентрация активатора становится равной 0. При наличии соседнего элемента, находящегося в возбужденном состоянии, она увеличивается на 1. Если p ближайших соседей возбуждены, то на соответствующем шаге к предыдущему значению концентрации активатора прибавляется число возбужденных соседей.

Program PROGRAMMA5;

uses dos, crt, graph; const N=110; M=90; s=13; r=6; h=5;

var y, yy, u : array [1..N,1..M] of integer; ii, jj, j, k, Gd, Gm : integer; i : Longint; Label met;

BEGIN Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi'); If GraphResult <> grOk then Halt(1);

setcolor(8); setbkcolor(15); (* y[50,50]:=1; { Одиночная волна } *)

For j:=1 to 45 do For i:=1 to 13 do y[40+i,j]:=i; { Однорукавная волна }

(*For j:=1 to M do { Двурюкавная волна }

For i:=1 to 13 do begin y[40+i,j]:=i; If j>40 then y[40+i,j]:=14-i; end; *)

Repeat

If k=round(k/20)*20 then y[30,30]:=1; {Пейсмекер 1 }

(* If k=round(k/30)*30 then y[20,50]:=1; {Пейсмекер 2 } *)

For i:=2 to N-1 do For j:=2 to M-1 do begin

If (y[i,j]>0) and (y[i,j]<s) then yy[i,j]:=y[i,j]+1;

If y[i,j]=s then begin yy[i,j]:=0; u[i,j]:=0; end;

If y[i,j] <> 0 then goto met;

For ii:=i-1 to i+1 do For jj:=j-1 to j+1 do begin

If (y[ii,jj]>0) and (y[ii,jj]<=r) then u[i,j]:=u[i,j]+1;

If u[i,j]>=h then yy[i,j]:=1; end;

met: end; Delay(2000); {Задержка}

cleardevice;

For i:=21 to 70 do begin {Препятствие}

yy[i,60]:=0; yy[i,61]:=0; circle(6*i-10,500-6*60,3); circle(6*i-10,500-6*61,3); end;

For i:=1 to N do For j:=1 to M do

begin y[i,j]:=yy[i,j];

setcolor(12); If (y[i,j]>=1) and (y[i,j]<=r) then circle(6*i-10,500-6*j,3);

```
setcolor(8); If (y[i,j]>6) and (y[i,j]<=s) then circle(6*i-10,500-6*j,2);  
end;  
until KeyPressed; CloseGraph;  
END.
```

Рассмотренная компьютерная модель активной среды позволяет изучить целую совокупность автоволновых явлений (рис.7): распространение одиночного возмущения, образование автоволн, дифракция автоволн, возникновение однорукавной и двурукавной волн, синхронизация колебаний элементов среды, самоорганизация и другие.

Более подробный анализ перечисленных компьютерных моделей, с соответствующей теорией, алгоритмами, текстами задач, цветными рисунками представлен на сайте maier-gv.glazov.net.

ЛИТЕРАТУРА

1. Гулд Х., Тобочник Я. Компьютерное моделирование в физике: В 2-х частях. Часть первая. — М.: Мир, 1990.— 400 с.
2. Лоскутов А.Ю., Михайлов А.С. Введение в синергетику: Учеб. руководство.— М.: Наука, 1990.— 272 с.
3. Майер Р.В. Основы компьютерного моделирования: Учебн.пособие. — Глазов: ГГПИ, 2005. — 25 с.
3. Могилев А.В., Пак Н.И., Хеннер Е.К. Информатика: Учеб. Пособие для студ. пед.вузов / Под ред. Е.К.Хеннер.— М.: Изд.центр "Академия", 2001. — 816 с.
4. Тихонов А.Н., Самарский А.А. Уравнения математической физики.— М.: Наука, 1966.— 724 с.