Майер Роберт Валерьевич, robert_maier@mail.ru

РЕШЕНИЕ ВАРИАЦИОННЫХ ЗАДАЧ НА КОМПЬЮТЕРЕ

В задачах на вариационное исчисление ищут функцию (кривую, поверхность), для которой заданный функционал принимает экстремальное значение. Проанализируем способы решения вариационных задач на ПЭВМ.

Задача 1. ***Имеется неоднородная нить или цепь. Определите форму, которую она примет в однородном поле тяжести, если ее концы закрепить в фиксированных точках.*** Решение: Нить примет форму, при которой ее потенциальная энергия минимальна. Мысленно заменим нить совокупностью материальных точек с массами m_i, которые связаны друг с другом пружинками жесткостью k и длиной b. Расстояние между соседними частицами и потенциальная энергия всей системы определяются уравнениями:

$$l_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}, \qquad U = \sum_{i=1}^{N} (k(l_i - b)^2 / 2 + m_i g y_i).$$

Допустим, правый конец нити привязан математическому маятнику длиной c и массой m_N. Заменим нить маятника пружиной с большой жесткостью k и длиной b_1, тогда к потенциальной энергии системы следует прибавить $\Delta U = k(l_1 - b_1)^2 / 2$. В программе ПР – 1 перебираются материальные точки m_i, случайным образом изменяются их координаты и каждый раз вычисляется получающееся значение потенциальной энергии системы. Если при смещении i–ой частицы потенциальная энергия уменьшилась, то это новое состояние системы принимается, иначе — отвергается. Результат вычислений — на рис. 1.

```
{$N+}  uses  dos, crt, graph;  const  N=20; b=20; b1=100; k=400; g=10; pi=3.1415;       { ПР – 1 }
var U,U0,U1,x1,y1,l,l1,dU: Real; i,j,t,Gd,Gm: integer;                                  { Borland Pascal 7.0 }
x,y,m: array [1..N] of single; Energiya: string;
Procedure Energy;  var i: integer;
begin U:=0; For i:=2 to N do begin l:=sqrt(sqr(x[i]-x[i-1])+sqr(y[i]-y[i-1]));
  If l > b then dU:=k*sqr(l-b)/2; U:=U+dU+m[i]*g*y[i]; end;
  l1:=sqrt(sqr(x[N]-400)+sqr(y[N]-60)); U:=U+k*sqr(l1-b1)/2; end;
BEGIN Randomize; Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
For i:=1 to N do begin x[i]:=20*i; y[i]:=-2*i;
   If  i < N then m[i]:=2 else m[i]:=10; end;
Repeat inc(t);   For i:=2 to N do begin x1:=x[i]; y1:=y[i]; Energy; U0:=U;
     x[i]:=x[i]+random(100)/25-2; y[i]:=y[i]+random(100)/25-2; Energy; U1:=U;
     If  U1 > U0 then begin x[i]:=x1; y[i]:=y1; end;
     end; Energy; Str(U,Energiya);
  If  t mod 500=1 then begin cleardevice;
     For i:=1 to N-1 do begin circle(10+round(x[i]),150-round(y[i]),3);
     line(10+round(x[i]),150-round(y[i]),10+round(x[i+1]),150-round(y[i+1])); end;
     line(10+round(x[N]),150-round(y[N]),410,150-60); OuttextXY(10,450,Energiya); end;
until KeyPressed; CloseGraph;
END.
```

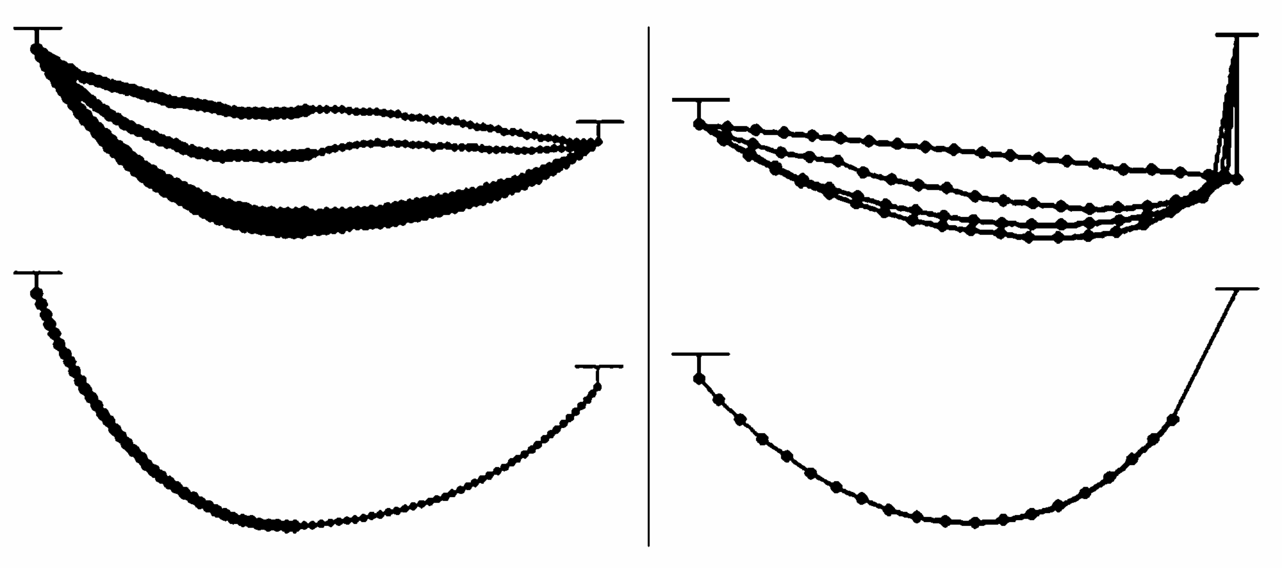

Рис. 1. Результаты вычисления формы нити.

<u>Задача 2.</u> Имеется неоднородная цепь, ее концы закреплены в фиксированных точках. К заданной точке цепи привязана невесомая нить, которая перекинута через неподвижный блок и соединена с грузом известной массы M. Требуется рассчитать форму нити. <u>Решение:</u> Заменим цепь совокупностью материальных точек, соединенных пружинами жесткостью k и длиной b. Пусть блок имеет небольшие размеры и его координаты равны X и Y, а перекинутая через него нить привязана к k–ой материальной точке с координатами x_k и y_k. Тогда при расчете потенциальной энергии системы следует учесть потенциальную энергию груза массы M: $\Delta U = Mg\sqrt{(x_k - X)^2 + (y_k - Y)^2} + C$. Если к некоторой i–ой точке цепи привязан груз известной массы M_1 (без блока), то при расчете формы цепи необходимо увеличить массу i–ой точки на величину M_1. Во всем остальном задача решается аналогично задаче 1 (программа ПР – 2). Результат решения приведен на рис. 2.

```
{$N+}  uses  dos, crt, graph; const  N=80; b=5; k=200; g=10; pi=3.1415;                    { ПР – 2 }
var U,U0,U1,x1,y1,l,dU,S,mm: real; i,j,t,r,Gd,Gm: integer;                            { Borland Pascal 7.0 }
Energiya: string;  x,y,m: array [1..N] of single;
Procedure Energy;
var i: integer;
begin U:=0; For i:=2 to N do begin l:=sqrt(sqr(x[i]-x[i-1])+sqr(y[i]-y[i-1]));
   If  l > b then dU:=k*sqr(l-b)/2; U:=U+dU+m[i]*g*y[i]; end;
   S:=sqrt(sqr(x[40]-200)+sqr(y[40]-20)); U:=U+mm*g*S; end;
BEGIN  Randomize; Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
For i:=1 to N do begin x[i]:=6*i; y[i]:=-i;
If  I < 40 then m[i]:=2 else m[i]:=4; m[20]:=30; end; mm:=25;
Repeat inc(t);
For i:=2 to N-1 do begin  x1:=x[i]; y1:=y[i]; Energy; U0:=U;
    x[i]:=x[i]+random(100)/25-2;  y[i]:=y[i]+random(100)/25-2; Energy; U1:=U;
    If  U1 > U0 then begin x[i]:=x1; y[i]:=y1; end; end; Energy; Str(U,Energiya);
If t mod 500=1 then begin cleardevice;
```

```
For i:=1 to N-1 do begin If  i < 40 then r:=1 else r:=4;
    circle(10+round(x[i]),150-round(y[i]),2); circle(10+round(x[i]),150-round(y[i]),r);
    line(10+round(x[i]),150-round(y[i]),10+round(x[i+1]),150-round(y[i+1])); end;
    line(10+round(x[40]),150-round(y[40]),200,20); OuttextXY(10,450,Energiya);  end;
until KeyPressed; CloseGraph;
END.
```

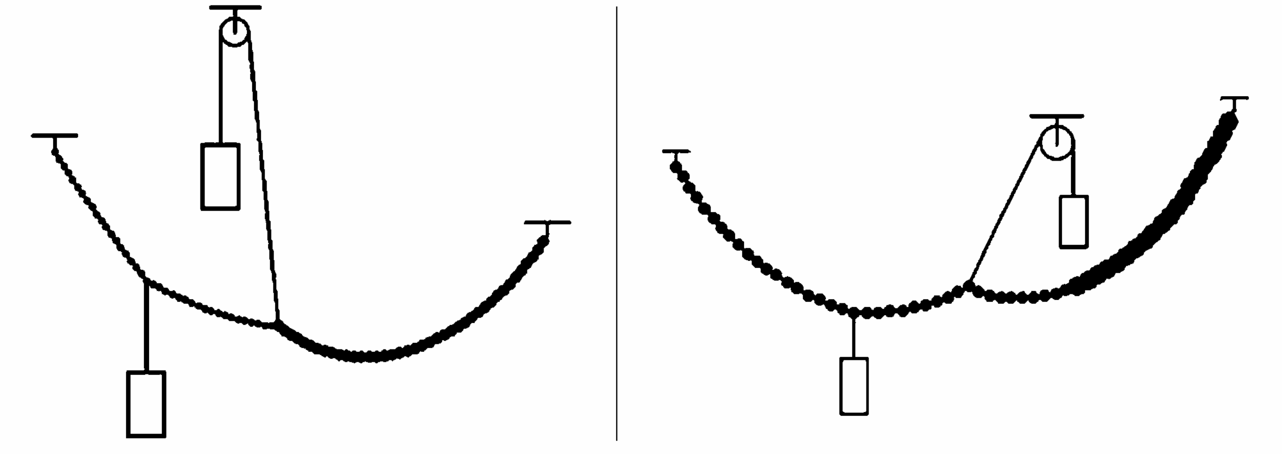

Рис. 2. Результат вычисления формы нити.

<u>***Задача 3.***</u> ***Рассчитайте форму упругой пластины, находящейся в однородном поле тяжести. Пластина неоднородная, один ее конец закреплен.*** <u>Решение:</u> Пластину представим как систему материальных точек m_i, связанных недеформируемыми стержнями длиной b. При изгибе пластины изменяются угол φ_i и координаты x_i, y_i. Потенциальная энергия системы:

$$U = \sum_{i=1}^{N} k_i(\varphi_i - \varphi_{i-1})^2 / 2 + m_i g y_i .$$

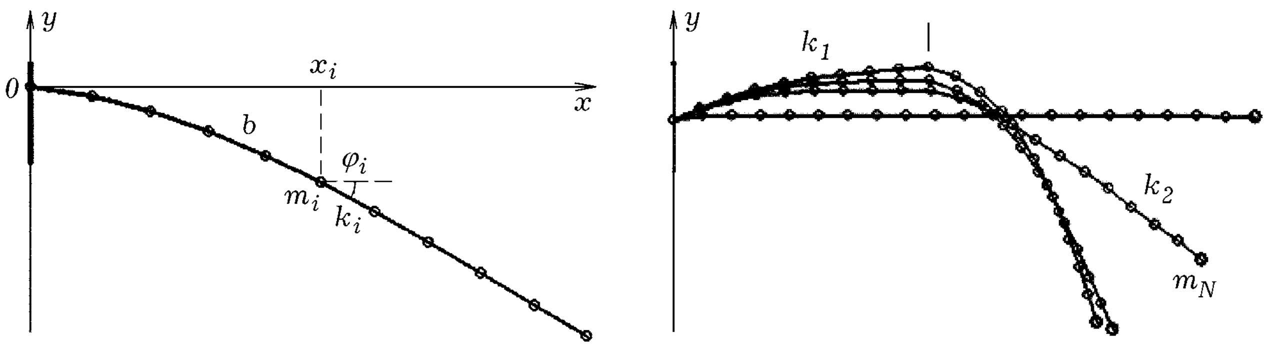

Рис. 3. К вычислению формы упругой пластины.

В используемой программе ПР – 3 реализуется следующий алгоритм. Последовательно перебираются материальные точки m_i и случайным образом изменяются углы φ_i. Каждый раз пересчитывается энергия системы. Если она увеличилась, то эта конфигурация отвергается, а если уменьшилась — принимается. В результате определяется устойчивое состояние равновесия системы, соответствующее минимуму потенциальной энергии. На рис. 3 представлены результаты расчетов для неоднородного стержня (жесткости левой и правой половин различны), к концу которого прикреплен груз (масса m[N] в 5 раз больше масс других материальных точек).

```
{$N+} uses  dos, crt, graph; const  N=20; b=20; g=10; pi=3.1415;                    { ПР – 3 }
var U,U0,U1,x1,y1,fi0,l,l1,dU: Real; i,j,t,Gd,Gm: integer;                  { Borland Pascal 7.0 }
fi,x,y,m,k: array [0..N] of single; Energiya: string;
Procedure Energy; var i: integer;
begin U:=0; For i:=1 to N do begin x[i]:=x[i-1]+b*cos(fi[i]); y[i]:=y[i-1]-b*sin(fi[i]); end;
  For i:=1 to N do begin dU:=k[i]*sqr(fi[i]-fi[i-1])/2; U:=U+dU+m[i]*g*y[i]; end; end;
BEGIN Randomize; Gd:= Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
For i:=0 to N do begin fi[i]:=0;
   If i < N-1 then m[i]:=0.02 else m[i]:=0.1;
   If i < N/2 then k[i]:=20000 else k[i]:=1000; end; fi[0]:=-0.5;
Repeat inc(t); For i:=N downto 1 do
    begin fi0:=fi[i]; Energy; U0:=U; fi[i]:=fi[i]+pi*(random(200)/100-1)/20;
      Energy; U1:=U; If U1 > U0 then fi[i]:=fi0;
    end; Energy; Str(U,Energiya);
  For i:=1 to N do begin x[i]:=x[i-1]+b*cos(fi[i]); y[i]:=y[i-1]-b*sin(fi[i]); end;
  If t mod 50=1 then begin cleardevice;
      For i:=0 to N do begin circle(20+round(x[i]),150-round(y[i]),3);
      line(20+round(x[i]),150-round(y[i]),20+round(x[i-1]),150-round(y[i-1])); end;
      circle(20+round(x[N]),150-round(y[N]),5);
      OuttextXY(10,450,Energiya); end;
until KeyPressed; CloseGraph;
END.
```

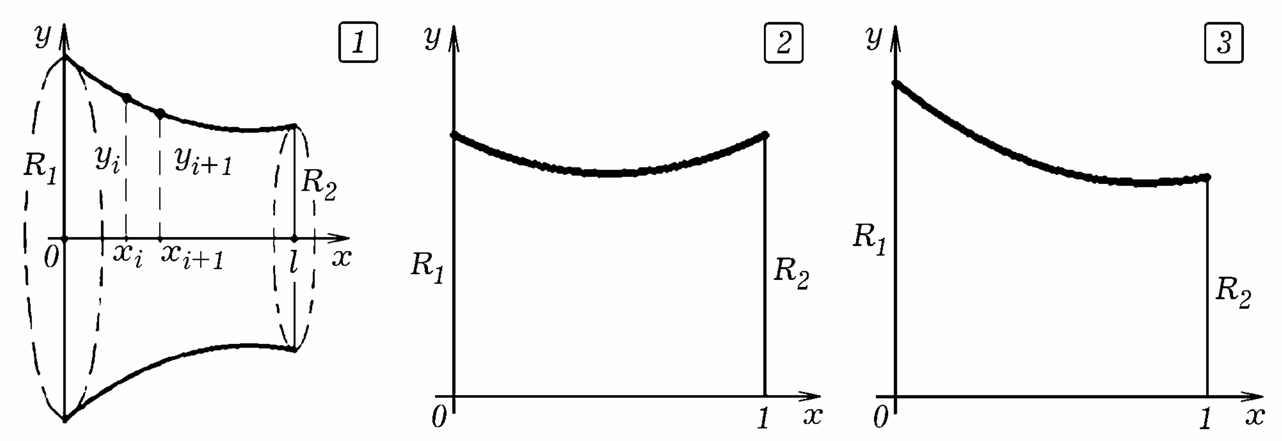

Рис. 4. Форма мыльной пленки, натянутой между двумя кольцами.

<u>Задача 4.</u> Имеются два кольца, центры которых лежат на одной оси, перпендикулярной содержащим их плоскостям. Рассчитайте форму натянутой на них мыльной пленки. <u>Решение:</u> Задача состоит в нахождении формы пленки, при которой ее потенциальная энергия, пропорциональная площади S, минимальна. Пленка примет осесимметричную форму, определяемую функцией $y = y(x)$ (рис. 4.1). Ее площадь поверхности

$$S = \pi \sum_{i=1}^{N} (y_i + y_{i+1}) \sqrt{\Delta x^2 + (y_{i+1} - y_i)^2} .$$

Используется программа ПР – 4. В процедуре Ploshad осуществляется вычисление площади S пленки. Затем перебираются все значения $y_i = y(x_i)$ и уменьшаются на 0.0001. При этом каждый раз вычисляется новое значение площади S1. Если площадь уменьшилась, то изменение $y_i = y(x_i)$ принимается, а если нет — отвергается. Все это многократно повторяется, результаты вычислений выводятся в виде графика на экран (рис. 4.2 и 4.3). При слишком малых R_1, R_2 и большом l задача не имеет решения.

```
{$N+}  uses dos, crt, graph; const N=100; R1=1; R2=0.7; pi=3.1415926;                    { ПР – 4 }
var l,S,dx,a,S0,S1: real; i,j,k,Gd, Gm: integer; y: array[1..N]of single;          { Borland Pascal 7.0 }
Procedure Ploshad; begin S:=0; For i:=1 to N-1 do
S:=S+pi*(y[i]+y[i+1])*sqrt(sqr(y[i]-y[i+1])+dx*dx); end;
BEGIN  S:=0; dx:=1/N; Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
For i:=1 to N do y[i]:=R1; y[N]:=R2;
Repeat inc(k);
  For j:=2 to N-1 do begin a:=y[j]; Ploshad; S0:=S; y[j]:=a-0.0001; Ploshad; S1:=S;
     If S1>S0 then y[j]:=a; end;
  If k mod 100=0 then begin cleardevice;  line(0,400,640,400);
    For i:=1 to 100 do  circle(20+round(300*i*dx),400-round(300*y[i]),2); end;
until KeyPressed; CloseGraph;
END.
```

<u>Задача 5.</u> Рассчитайте форму капли жидкости, лежащей на горизонтальной пластине в поле тяжести Земли. Жидкость частично смачивает пластину. <u>Решение:</u> На жидкость действуют силы тяжести и поверхностного натяжения. Капля принимает форму, при которой ее потенциальная энергия минимальна. Форму капли будем аппроксимировать эллипсоидом вращения вокруг вертикальной оси, нижняя часть которого срезана (рис. 5). Уравнение осевого сечения капли имеет вид: $x^2/a^2 + (y-c)^2/b^2 = 1$, где $y > 0$. Площадь всей поверхности капли, включая площадь соприкосновения с пластиной, и ее потенциальная энергия равны:

$$S = \pi \sum_{i=1}^{N} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}\,(x_{i+1} + x_i) + K_1 \pi r^2,$$

$$U = \rho g \pi \sum_{i=1}^{N} \left(\frac{x_{i+1} + x_i}{2} \right)^2 \cdot (y_{i+1} - y_i) \cdot \frac{y_{i+1} + y_i}{2} + K_2 S.$$

В программе изменяются параметры a и c, а b вычисляется так, чтобы объем капли оставался неизменным. Принимаются такие значения a, b и c, при которых потенциальная энергия системы минимальна. Меняя K_1 и K_2, можно рассчитать форму капли жидкости при различных плотностях, коэффициентах поверхностного натяжения и смачивании поверхности (рис. 5).

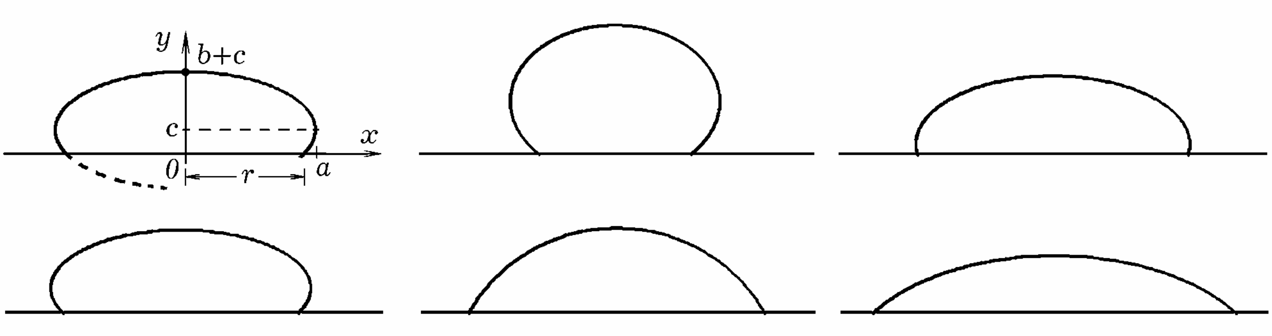

Рис. 5. Результаты расчета формы капли.

```
{$N+} uses crt, graph;                                                    { ПР – 5 }
var k,i, j,DV, MV : integer; xx,yy,x,y : array[0..205] of extended;       { Borland Pascal 7.0 }
r,a1,b1,c1,a,b,c,U,S,V,U1,V0 : real;
procedure Energy; var i:integer;
begin S:=0; V:=0; U:=0; For i:=0 to 200 do begin
y[i]:=c+b-b*i/100; x[i]:=a*sqrt(abs(1-(c-y[i])*(c-y[i])/b/b));end;
For i:=0 to 199 do if y[i]>=0 then begin
S:=S+3.14*sqrt(sqr(x[i]-x[i+1])+sqr(y[i]-y[i+1]))*(x[i]+x[i+1]);
V:=V+3.14*sqr(x[i]+x[i+1])*abs(y[i]-y[i+1])/4;
U:=U+3.14*sqr(x[i]+x[i+1])*(y[i]-y[i+1])*(y[i]+y[i+1])/8;
r:=x[i]; end; S:=S+0.8*3.14*r*r; U:=U+1.2*S; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
randomize; a:=1; b:=1; c:=b; Energy; U1:=U; V0:=V;
repeat Energy; a1:=a; b1:=b; c1:=c;
a:=a+0.3*random(100)/100; c:=c+0.3*random(100)/100-0.1; b:=0.1;
Repeat b:=b+0.005; Energy; until V>V0; Energy; k:=0;
if (U>U1) then begin a:=a1; b:=b1; c:=c1; k:=1; end;
if (U<U1) then U1:=U; if k=0 then begin cleardevice;
For i:=0 to 199 do begin y[i]:=c+b-b*i/100; x[i]:=a*sqrt(abs(1-(c-y[i])*(c-y[i])/b/b));
if y[i]>=0 then begin
line(320+round(120*x[i]),300-round(120*y[i]),320+round(120*x[i+1]),300-round(120*y[i+1]));
line(320-round(120*x[i]),300-round(120*y[i]),320-round(120*x[i+1]),300-round(120*y[i+1]));
end; end; line(0,300,640,300); end;
until KeyPressed; CloseGraph;
END.
```

<u>Задача 6.</u> Показатель преломления неоднородной среды зависит от координат: $n = n(x, y)$. ***Рассчитайте траекторию распространения светового луча из точки А в точку В.*** <u>Решение:</u> По принципу Ферма, свет распространяется по пути, оптическая длина которого минимальна (экстремальна). Рассмотрим сетку, состоящую из вертикалей, пересекающих ось абсцисс в точках $x_i = i\Delta x$, $i = 0,1,...,N$. Траекторию аппроксимируют ломаной, пересекающей линии сетки в точках $y_1 = y(x_1)$, $y_2 = y(x_2)$, ..., $y_N = y(x_N)$. Оптическая длина пути равна:

$$L = \sum_{i=1}^{N} n_{i,j} \sqrt{\Delta x^2 + (y_{i-1} - y_i)^2} ,$$

где $n_{i,j}$ — показатель преломления в точках с координатами x_i и y_i. Будем случайным образом варьировать величины y_i, каждый раз определяя изменение оптической длины пути L. Если вариация y_i вызывает уменьшение оптической длины пути, то она принимается, а в противном случае, — отвергается. Используется программа ПР – 7; результаты — на рис.7.

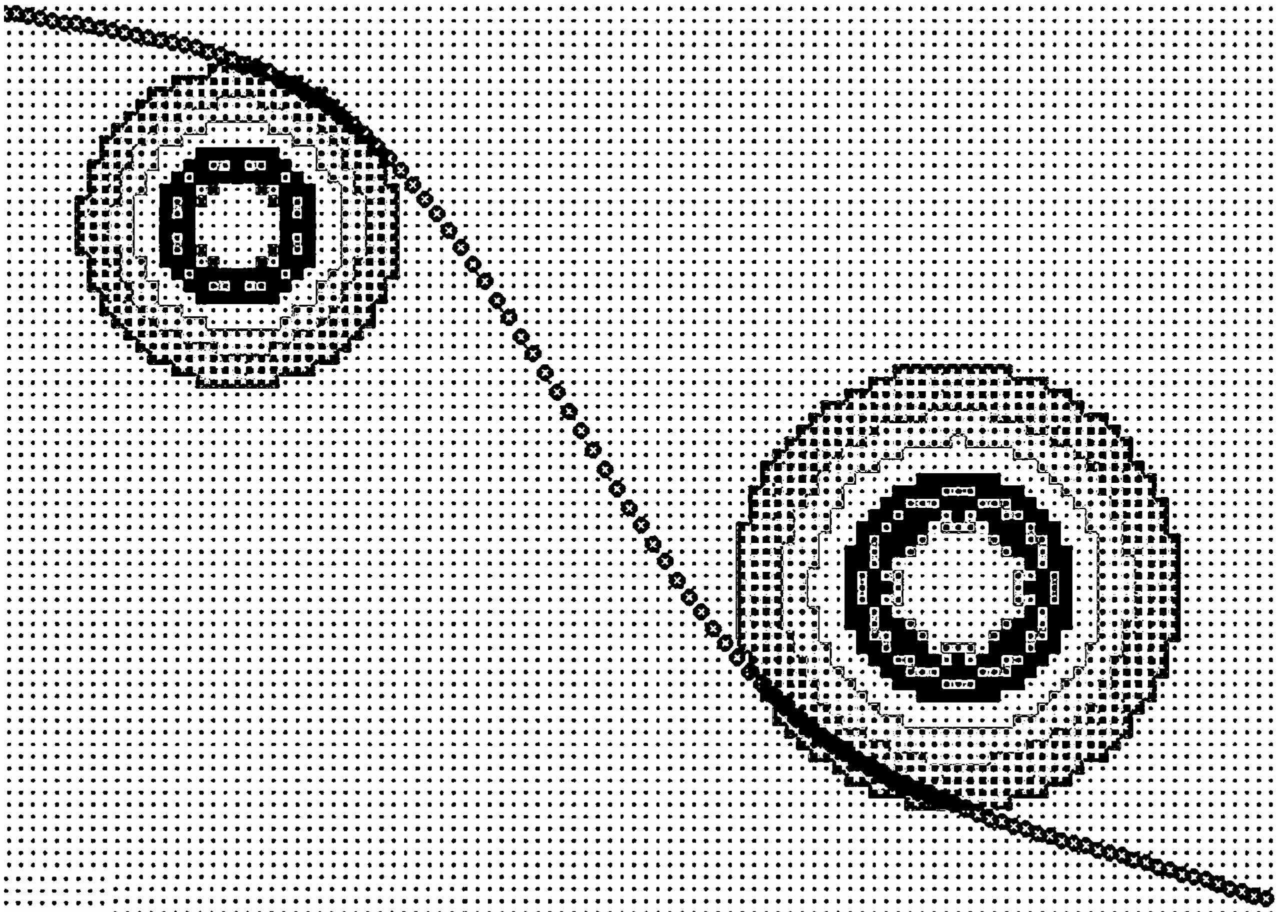

Рис 7. Траектория распространения луча в неоднородной среде.

```
{$N+} uses dos, crt, graph;                                              { ПР – 6 }
const N=106; M=80; R1=1; R2=0.7; pi=3.1415926;                   { Borland Pascal 7.0 }
var s,s1,l,dx,a,xx,yy,L0,L1,n_sr: real; x,j,k,Gd, Gm: integer;
y: array[1..N] of single; Dlina: string;
Procedure Opt_plotn;
begin s:=sqr(20-xx)+sqr(20-yy); s1:=sqr(80-xx)+sqr(50-yy);
  n_sr:=1+50/(s+1)+100/(s1+1);  If n_sr>5 then n_sr:=5; end;
Procedure Opt_Dlina; var x:integer;
begin L:=0; For x:=1 to N-1 do begin xx:=x; yy:=y[x]; Opt_plotn;
L:=L+n_sr*sqrt(sqr(y[x]-y[x+1])+dx*dx); end; end;
BEGIN Randomize; dx:=1; Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
For x:=1 to N do y[x]:=2+0.7*x;
Repeat inc(k); For x:=2 to N-1 do begin a:=y[x]; Opt_Dlina;
L0:=L; y[x]:=a+random(20)/100-0.1; Opt_Dlina; L1:=L;
If L1>L0 then y[x]:=a; end; Opt_Dlina; Str(L,Dlina);
If k mod 100=0 then begin cleardevice;
For x:=1 to N do For j:=1 to M do begin xx:=x; yy:=j; Opt_plotn;
setcolor(round(n_sr*5)+5); rectangle(20+5*x,20+5*j,24+5*x,24+5*j);
```

```
rectangle(21+5*x,21+5*j,23+5*x,23+5*j); end;
OuttextXY(20,440,Dlina); setcolor(8);
For x:=1 to N do begin circle(20+round(5*x),20+round(5*y[x]),1);
circle(20+round(5*x),20+round(5*y[x]),3); end; end;
until KeyPressed; CloseGraph;
END.
```

Рассмотренные задачи и их решения могут быть использованы при изучении курсов “Численные методы” и “Компьютерное моделирование”. Автор рекомендует обратиться к электронному пособию “Задачи, алгоритмы, программы”, которое можно скачать с сайта http://maier-rv.glazov.net.

Примечания

1. Майер Р.В. Компьютерное моделирование физических явлений: Монография. — Глазов: ГГПИ, 2009. — 111 с. (http://maier-rv.glazov.net)
2. Майер Р.В. Задачи, алгоритмы, программы: Учебное пособие [Электронный ресурс]. — Глазов, 2010. (http://komp-model.narod.ru).
3. Самарский А.А., Вабищевич П.Н., Самарская Е.А. Задачи и упражнения по численным методам: Учеб. пособие. — М.: Эдиторал УРСС, 2000. — 208 с.