

ДОДАТОК А

Системний аналіз поставленої проблеми

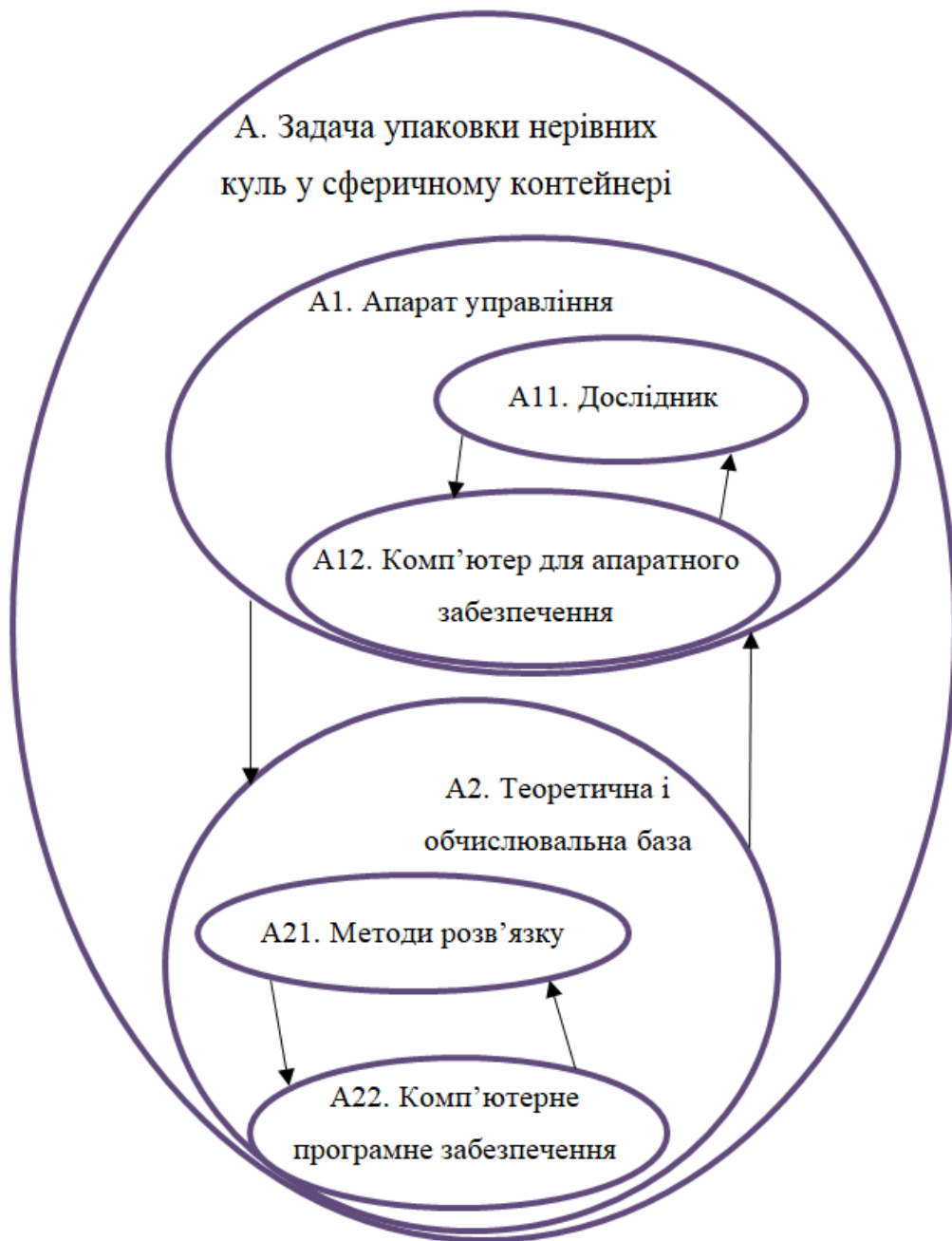


Рисунок А.1 – Модель типу «білий ящик»



Рисунок А.2 – Контекстна діаграма



Рисунок А.3 – Основні функції системи: рівень A0

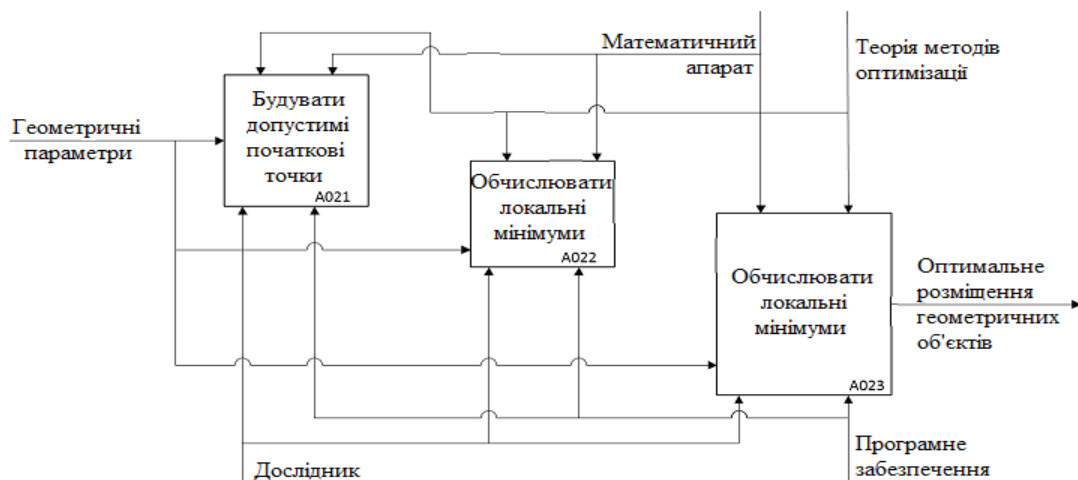


Рисунок А.4 – Основні функції системи: рівень A2

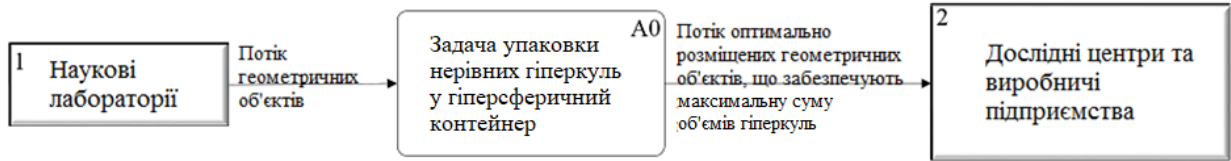


Рисунок А.5 – DFD-діаграма

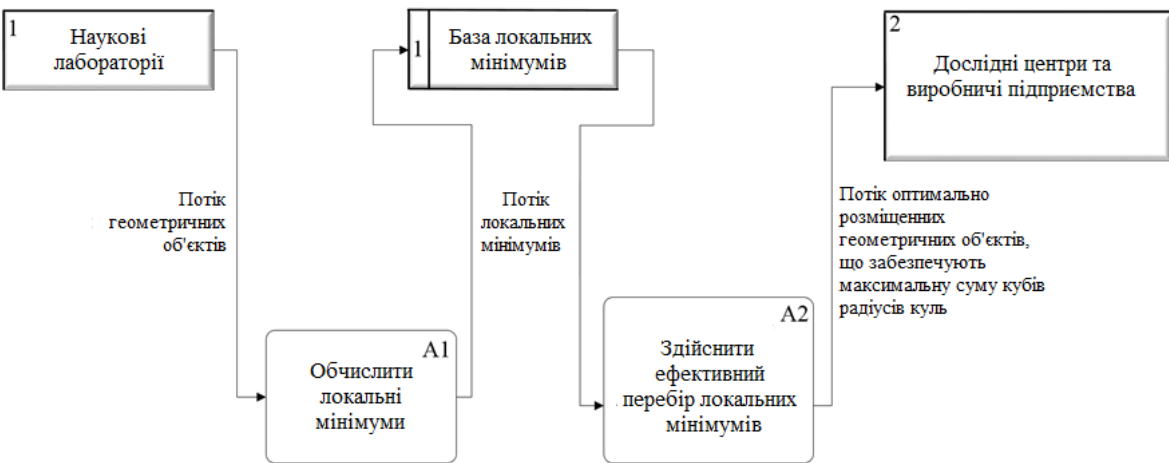


Рисунок А.6 –DFD-діаграма 1-го рівня декомпозиції

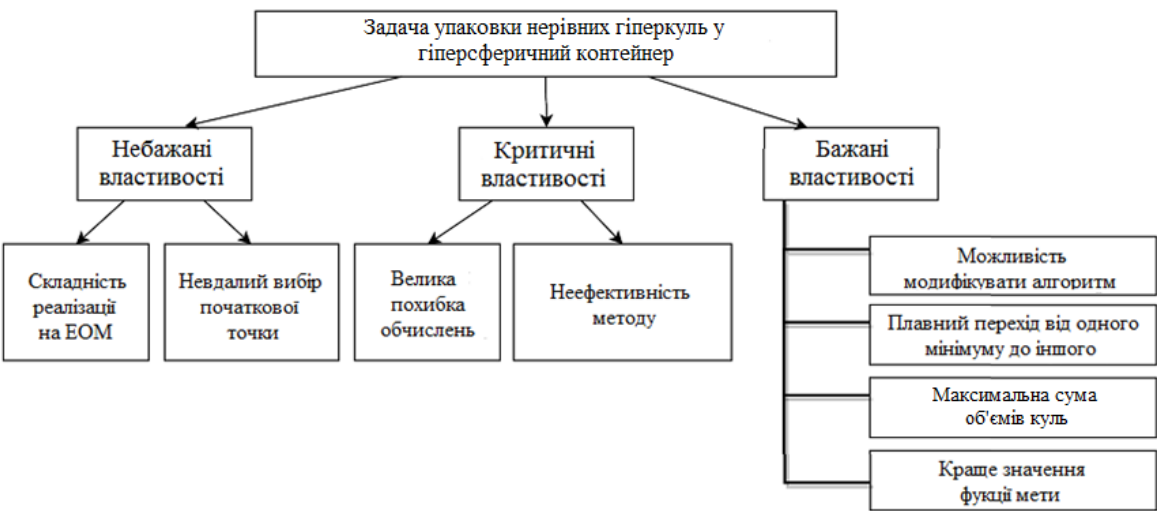


Рисунок А.7 – Ієрархічна модель аналізу незадоволенностей

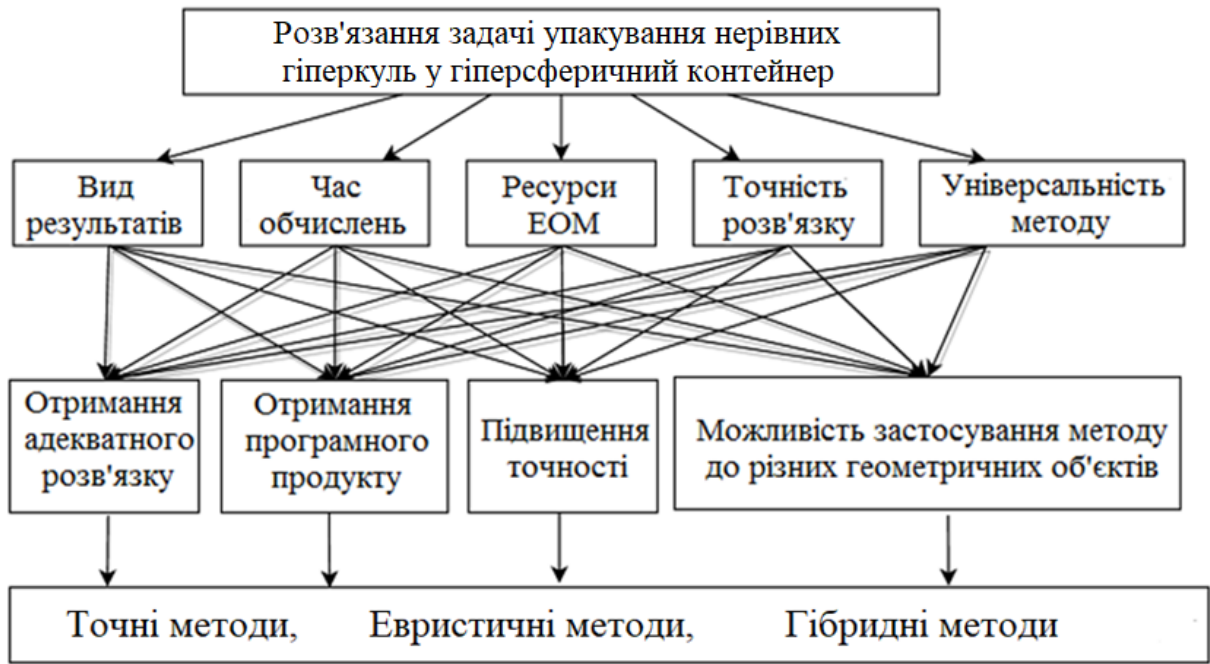


Рисунок А.8 – Змістовна ієрархічна модель
прямого процесу аналізу проблеми

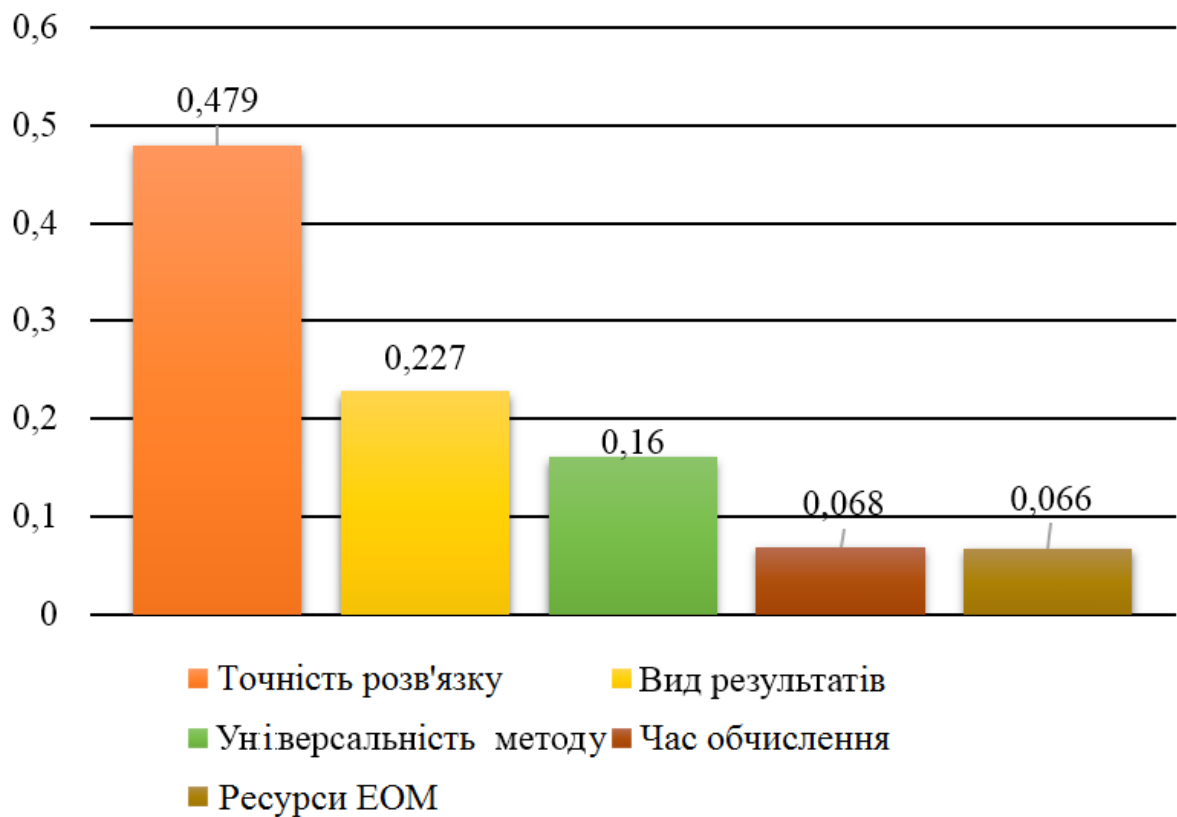


Рисунок А.9 – Діаграма пріоритетів для нульового рівня

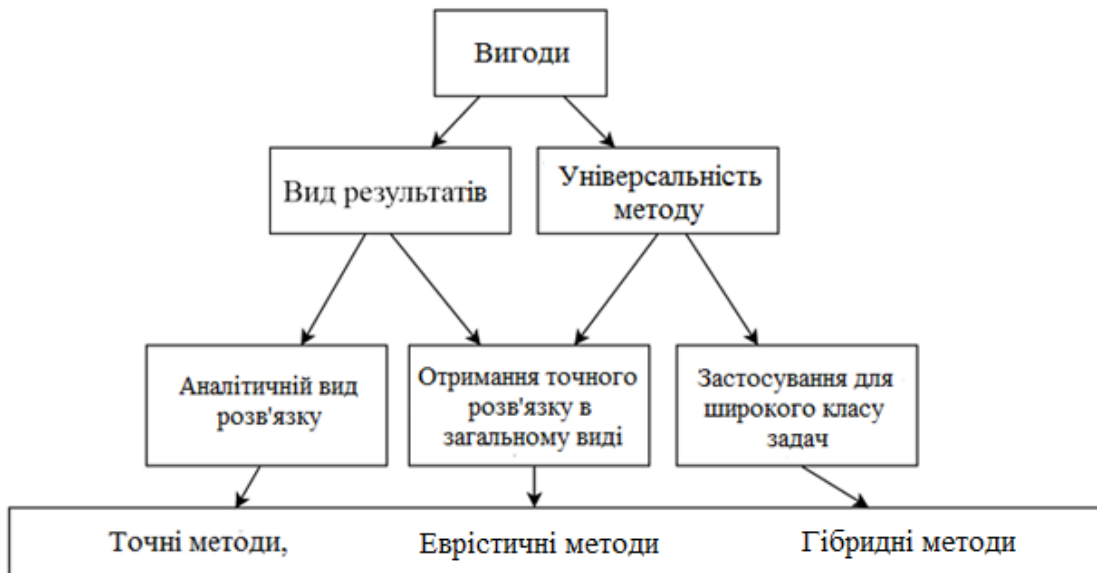


Рисунок А.10 – Змістовна модель оцінки вектора пріоритетів за критерієм «вигоди»

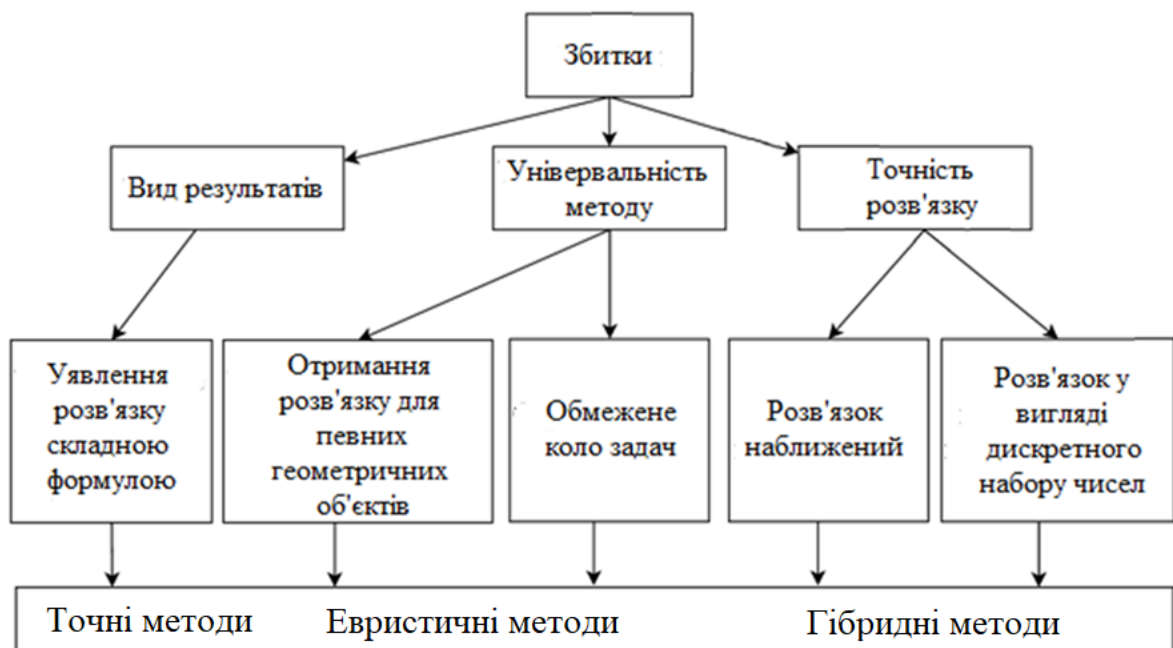


Рисунок А.11 – Змістовна модель оцінки вектора пріоритетів за критерієм «збитки»

ДОДАТОК Б

Лістинг програми

```

unit Unit1 copy;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    PaintBox1: TPaintBox;
    procedure Button1Click(Sender:
      TObject);
    procedure Button2Click(Sender:
      TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const eps1=1e-10;nmax=600;eps=1e-
9;eps2=1e-4;ninit=20;
acur=1e-8;
  ia=50000;
  nmax5=15000;
  epsb=1e-14;small=1e-18;
{BPMPD}

  const maxm= 10000;
  const maxn= 10000;
  const maxnz= 50000;
  const maxqnz=1;
  type tnz =array [1..maxnz] of integer;
  type tqnz =array [1..maxqnz] of integer;
  type tn =array [1..maxn] of integer;
  type tnn =array [1..maxm+maxn] of
    integer;
  type tdnz =array [1..maxnz] of double;
  type tdqnz=array [1..maxqnz] of double;
  type tdm =array [1..maxm] of double;
  type tdn =array [1..maxn] of double;
  type tdmn =array [1..maxm+maxn] of
    double;
{BPMPD}
const tochn=1e-4;
type tr2=array[1..2] of double;
sin5=array[0..nmax5] of integer;
lnr5=array[0..nmax5] of boolean;
tiar=array[1..IA] of double;
tai=array[1..IA] of integer;
trn5=array[0..nmax5] of double;
trn=array[1..nmax] of double;
tbn=array[1..nmax] of byte;
tdn=array[-5..nmax..100..nmax] of
boolean;
tr=array[0..nmax] of double;
{ tr1=array[1..nmax] of double;}
tni=array[0..nmax] of integer;
t2ni=array[0..500] of integer;
t2ri=array[0..500] of double;
tr200=array[1..500] of double;
ti200=array[1..500] of integer;
type tr4=array[1..2] of double;
tr5=array[0..5] of double;

  t_p_trn='^trn;
  t_p_xfull='^tdm;

var Form1: TForm1;
  MyPaintBox: TPaintBox;
  {graphic} fposl2:text;
  {
  ass,bs,cs,x,miny,abkv,miy,miy1,alfa,bett
  a,nch,sqrmch,

sqrmchf,nchf,bettaf,bettaa,ncha,sqmcha,r
omem,ndel,kest,ndouble,ss,ss1,

mo1max,zcurr,zcurra,delh_mo_2,sigma:d
ouble;
  }
  {endgraph}
  FName:String;
  ipar,jpar:tnz;
  rpar:tdnz;
  iparami,nparami,isin,icou,tofp
integer;
  xfull:tdm;
  { h,m,hund,s3,h1,m1,s31:word;}
  dtime:longint;
  llll,lprob,lprob2,lror:boolean;
  c:char;
  xex,yex:tr5;
  nex,idelmax,iglcount,ilid:integer;

  zest4min,zest5min,zest1min,zest2min,le
ngmin,widx,widy,widz,xcyl,ycyl,xsph,ys
ph,zsph,rcyl,ztreebest:double;
var xan,yan,ran:tr4;
  rr1,xx,yy:array[1..30,1..30] of double;
  nr,ix,ix1,n,nixn,i:integer;

  rs,r,x,y,z,xbr,ybr,xbeg,ybeg,zbeg,rbeg,ri
n,rin1,rin2,xb,yb,rb,rvi,rvh,rvlg,rvhg,rh,x
h,yh,rph,mpl,phh,rpl,rvlh,rvh,rvt,delr,
xbddd,

ybddd,rbddd,zbddd,xbddd1,ybddd1,rbddd
1,r1ow,rhigh,xtreeb,ytreeb,ztreeb,rtreeb:
trn;

  xmem,ymem,zmem,rmem,xbest,ybest,zb
est,rbest,xmemdel,ymemdel,zmemdel,m
emdel,xbdut,ybdut,zbdut,rbdut,xbdut1,y
bdut1,rbdut1,

xbloc,ybloc,rbloc,zbloc,xbros,ybros,zbro
s,rbros,rinbros:trn;

  mp,nofvr,mpm,mpl,mphi,mpli,mprin,mpr
:tni;

  a0,a01,b0,zest,zestmin,zestdd,rmax,rmi
n,rmin1,mm,rpm,xxx,yyy,rr,rmaxest,r
maxi,rpm1,zestb,lemb:double;
  scale,zbestz,delmax:double;
  x1r,x2r,rsr:tr200;
  imser,nnm,iby,i8,nact,nactl:integer;
  lbeg,lx:boolean;
  fff,flo,fff1,fgr:text;

  g1,g2,ii,jj,j,nmk,nmk1,io,jo,ii,jp,jnr,luk,
ipa,jpa:integer;

  imk,jmk,imk1,jmk1,iim,igm,jgm,kmk,kg
m,iopmtm,jopmtm,kopmtm,iiopmtm:t2n
i;

  mdm,rnk1,{xmk1,ymk1,}rmk,xmk,ymk
:t2ri;

  LENL,LENU,NCP,LROW,LCOL:intege
r;
  iw11,iw12:tiai;
  lkod,ll,lkod2:boolean;
  {IP1,IP2:sin5;}
  IND1,IND2,IND1a,IND2a:tiai;
  A,aa:tia;
  W,B,ba:tm5;

  IW1,IW2,IW3,IW4,IW5,IW6,IW7,IW8:
sin5;

G,U,rat,rab,nko,sqrin,sqrr,rrab,zestbeg:
double;
  mpi,irand:integer;
var mc:tdnn;
  ftest,f1:text;
  n2,i1,icf,n3,idut,jdut:integer;
  m1,m2,or1,or2:sin5;
  igl:longint;
  it,proxod,itobl:integer;

  lr,lr1,lres,lrazr,lr1,lrh,lrhg,lr1,lrj,lrinj,l
rll,lrhh,lrhhh,lrhhh1,lrhhddd,lrll,lmem
del:lnr5;
  lty,ltoib:boolean;
  Nrestr,DimmatrB3:integer;
  xtarg,dt,kaux,xtargin,dt1,dtr:tm5;

  min,ss,svch,wtarg,wtarg2,wtarg0,wtarg
m,rr,zestb1,dis,ha,hb,xrab,yrab,zrab:dou
ble;
  idel,iadd,ismpg:integer;
  code,res,lkj,sedl:boolean;
  hc:array[1..nmax5] of double;
  nz,nza:integer;
  var lm:lnr5;
  mn:real,mnoz:tm5;
  nofvio,nofviol,k:integer;

  iv,jv,iv1,jv1,kv,kv1,iact,jact,iactl,jactl:ta
i;
  ikr:integer;
  nachrj1,nachnp2,nimi,nimj:integer;
  nresh,nlim,nsd,nreshbr,nreshmax:integer
;
  xt,yt,nach,kon,jnx,iny,delit,rj,sum,sumku
b,sumbegin,sumr:double;
  tol,tolcyl,deltax,deltay,rab1:double;
  lresh:boolean;
  sm1,sm2,jmin,iglob,irpl:integer;
  lhc,lparami:boolean;
  nofnegh:array[1..nmax] of integer;
  mofn:array[1..nmax,1..30] of integer;
  iper,jper,dper:integer;
  xper,yper,rper:array[1..ninit,1..nmax] of
double;
  sm1m,sm2m,summax:array[1..nmax] of
double;
  kappa,kappab,sqrdel,sqrmin,sqrdelc:dou
ble;
{BPMPD}
var
  {m,n,nz,}qn,qnz,{i,}mems,nbp:integer;
  var acolix:tnz;
  var qcolix:tdqnz;
  var acolcnt,qcolcnt,ukaz:tn;
  var acolnz:tdnz;
  var qcolnz:tdqnz;
  var rhs:tdm;
  var obj:tdn;
  var lbound,ubound,primal,dual:tdmn;
  var status:tn;
  var opt,big,mult:double;
  var codebp:integer;
  { type tnz =array [1..maxnz] of integer;
  type tqnz =array [1..maxqnz] of integer;
  type tn =array [1..maxn] of integer;
  type tnn =array [1..maxm+maxn] of
integer;
  type tdnz =array [1..maxnz] of double;
  type tdqnz=array [1..maxqnz] of double;
  type tdm =array [1..maxm] of double;
  type tdn =array [1..maxn] of double;
  type tdmn =array [1..maxm+maxn] of
double;
  }
  {inhib
  }

```



```

MoveTo(20,MyPaintbox.Height-50-
round(b0/scale));
lineTo(round(leng/scale)+20,MyPaintbo
x.Height-50-round(b0/scale));
MoveTo(20,MyPaintbox.Height-50);
lineTo(20,MyPaintbox.Height-50-
round(b0/scale));
MoveTo(20,MyPaintbox.Height-50);
lineTo(round(leng/scale)+20,MyPaintbo
x.Height-50);
MoveTo(round(leng/scale)+20,MyPaintb
ox.Height-50);
lineTo(round(leng/scale)+20,MyPaintbo
x.Height-50-round(b0/scale));
end;

end;
end;

procedure showdir(mn:real);
var i,ix,ix1,iy1,ix2,iy2:integer;
begin
MyPaintbox.Canvas.Pen.Color:=(CLred);
;
for i:=2 to n2+1 do
begin
if (i-1) mod 2 = 1 then
begin
ix1:=round((x[(i-1) div 2+1]/scale)+20;
ix2:=round((x[(i-1) div
2+1]+dt[i])/scale)+20;
end
else
begin
iy1:=MyPaintbox.Height-50-round((y[(i-1) div 2]/scale);
iy2:=MyPaintbox.Height-50-round((y[(i-1) div 2]+dt[i])/scale);
with MyPaintBox.Canvas do begin
MoveTo(ix1,iy1);
lineTo(ix2,iy2);
end;

end;
end;
MyPaintbox.Canvas.Pen.Color:=(CLblack);
end;

procedure
showcir(xc,yc,rc,rinc:double;ix,mpix,col
:integer);
var ix1,iy1,ix2,iy2,i:integer; s,s1:string;
begin
if ix=0 then
begin
with MyPaintBox.Canvas do begin
str(zest:9:5,s); s:=zest+'s+' tol=';
str(tol:8:6,s1);
s:=s+s1;
MyPaintbox.Canvas.TextOut(10,MyPain
tbox.Height-20,s);
end;

end;

ix1:=round((xc)/scale)+20;
iy1:=MyPaintbox.Height-50-
round((yc)/scale);

begin
{ if rc>rinc then}
{
MyPaintbox.Canvas.Pen.Color:=(clblack
);}
begin
if col=0 then
MyPaintbox.Canvas.Pen.Color:=(CLwhi
te);

```

```

if col=1 then
MyPaintbox.Canvas.Pen.Color:=(CLblac
k);
if col=8 then
MyPaintbox.Canvas.Pen.Color:=(CLgra
y);
if col=4 then
MyPaintbox.Canvas.Pen.Color:=(CLgre
en);
if col=3 then
MyPaintbox.Canvas.Pen.Color:=(CLblu
e);
end;
MyPaintbox.Canvas.Pen.Color:=col;
iy2:=trunc(rinc/scale)+1;
ix2:=trunc(rinc/scale)+1;

MyPaintbox.Canvas.Pen.Style:=psSolid;
if (rc<rinc) and (abs(rc-rinc)>1e-9) then
MyPaintbox.Canvas.Pen.Color:=(clred);
{if (rc>rinc) and (abs(rc-rinc)>1e-2)
then
MyPaintbox.Canvas.Pen.Color:=(clwhite
);}
{if (rc<rinc) and (abs(rc-rinc)>1e-9)
then
MyPaintbox.Canvas.Pen.Style:=psDash;
if (rc>rinc) and (abs(rc-rinc)>1e-2) then
MyPaintbox.Canvas.Pen.Style:=psDash;
}
MyPaintBox.Canvas.Ellipse(ix1-
ix2,iy1-iy2,ix1+ix2,iy1+iy2);
iy2:=trunc(rc/scale)+1;
ix2:=trunc(rc/scale)+1;

MyPaintbox.Canvas.brush.style:=bsClea
r;
if (abs(rc-rinc)>5e-2) then
begin
MyPaintbox.Canvas.Pen.Color:=col;
MyPaintBox.Canvas.Ellipse(ix1-
ix2,iy1-iy2,ix1+ix2,iy1+iy2);
end;

MyPaintbox.Canvas.Pen.Color:=col{((C
Lblack));

(*if (rc<rinc) and (abs(rc-rinc)>1e-7)
then
MyPaintbox.Canvas.Pen.Color:=(clred);
*)
if rc>rinc-tochn then
MyPaintBox.Canvas.Ellipse(ix1-
ix2,iy1-iy2,ix1+ix2,iy1+iy2);
{if rc>rinc then}
{MyPaintbox.Canvas.Pen.Color:=(CLbl
ack);}
begin
if col=0 then
MyPaintbox.Canvas.Pen.Color:=(CLwhi
te);
if col=1 then
MyPaintbox.Canvas.Pen.Color:=(CLblac
k);
if col=8 then
MyPaintbox.Canvas.Pen.Color:=(CLgra
y);
if col=4 then
MyPaintbox.Canvas.Pen.Color:=(CLgre
en);
if col=3 then
MyPaintbox.Canvas.Pen.Color:=(CLblu
e);
end;
end;

{str(mpix,s);
MyPaintbox.Canvas.TextOut(ix1+2,iy1+
2,s);}

```

```

str(ix,s);
for i:=1 to 20 do
MyPaintbox.Canvas.TextOut(ix1-12,iy1-
12,s);
ix1:=round((xc+a0)/scale)+20;
iy1:=MyPaintbox.Height-50-
round((yc+b0)/scale);
{setlinestyle(0,0,3);}
{line(ix1,iy1,ix1,iy1);}
if rc>1-tochn then;
{MyPaintBox.Canvas.Ellipse(ix1-1,iy1-
1,ix1+1,iy1+1);}

{setlinestyle(0,0,1);}
end;

procedure graph2d;
var i:integer;
begin
MyPaintBox.Refresh;
patt(a0,b0);
for i:=1 to n do
showcir(x[i],y[i],r[i],rin[i],i,i,1);
end;

procedure serv;
var s3,s1,s2,s4:string;
begin
begin
{ setfillstyle(1,white);
bar(0,getmaxy-10,getmaxx-
21,getmaxy);}
{bar(getmaxx-135,1,getmaxx,30);}
{ setcolor(black);}

{ str(r[n]:12:8,s1);str(n:2,s2);
outtextxy(10,getmaxy-
10,r['+s2+']+'+s1);}
str(r[itobl]:10:6,s1);
str(itobl:2,s4);
str(rin[itobl]:10:6,s2);
str(usage:5:2,s3);
MyPaintbox.Canvas.TextOut(2,MyPaint
Box.Height-30,r['+s4+']+'+s1+'--
'+s2+' Usage: '+s3+'%.');
end

end;

procedure convio(var nofvio:integer; var
iv,jv:tiai; var dis:double;
dopusk:double;
var i,j,k:integer; rab1,sumr:double;
log1:boolean;
begin
{icon:=icon+1;}
nofvio:=0; dis:=0;

for i:=1 to n do
begin
lrhh[i]:=false;
lrll[i]:=false;
end;
{if lprob then}
for i:=1 to n do
begin
{пересечение с зоной запрета -
цилиндром}

rab1:={sqrt}(sqrt((x[i]-
s0xq)*cosgamq*costetq+(y[i]-
s0yq)*cosgamq*(sintetq)-(z[i]-
s0zq)*(singamq))
+sqrt((x[i]-s0xq)*(sintetq)-(y[i]-
s0yq)*costetq))-sqrt(rcylq+r[i]));

```

```

if (rab1<0)and(abs(rab1)>dopusk) then
  begin
    nofvio:=nofvio+1;
    iv[nofvio]:=i;
    jv[nofvio]:=1;
    if abs(rab1)>dis then
      dis:=abs(rab1);
    end;
    {пересечение с зоной запрета -
    призмой}
    for m:=1 to mv do
      begin
        t1x[m]:=ee[m]+r[i]*anml[m-1];
        t2x[m]:=ee[m]+r[i]*bnml[m];
        t1y[m]:=ee[m]+r[i]*bnml[m-1];
        t2y[m]:=ee[m]+r[i]*bnml[m];
        t1z[m]:=ee[m]+r[i]*cnml[m-1];
        t2z[m]:=ee[m]+r[i]*cnml[m];
        end; {m}

    for m:=1 to mv do
      begin
        att[m]:=(t2y[m]-t1y[m])*wz-
        (t2z[m]-t1z[m])*wy;
        btt[m]:=(t2x[m]-
        t1x[m])*wz+(t2z[m]-t1z[m])*wx;
        ctt[m]:=(t2x[m]-t1x[m])*wy-
        (t2y[m]-t1y[m])*wx;
        dtt[m]:=t1x[m]*att[m]-
        t1y[m]*btt[m]-t1z[m]*ctt[m];

        gam2:=att[m]*x[i]+btt[m]*y[i]+ctt[m]*z
        [i]+dtt[m]*r[i];
        gam1:=sqrt(sqr((x[i]-
        ee[m])*cosgamz*cosctet+(y[i]-
        ff[m])*cosgamz*sintet-(z[i]-
        hh[m])*singamz)+
        sqr((x[i]-ee[m])*sintet-(y[i]-
        ff[m])*costet)-r[i];

        psi2sp[m]:=gam1;
        if gam2<gam1 then
          psi2sp[m]:=gam2;

        psi1sp[m]:=anml[m]*x[i]+bnml[m]*y[i]
        +cnml[m]*z[i]+dmml[m]*r[i];
        end; {m}
        mmax:=0; psi2:=1e+37;
        for m:=1 to mv do
          begin
            if psi2sp[m]>psi2 then
              begin
                psi2:=psi2sp[m]; mmax:=m;
              end;
            if psi1sp[m]>psi2 then
              begin
                psi2:=psi1sp[m]; mmax:=m;
              end;
            lmax:=1;
            rab1:=psi2;
            end; {m}
            if (rab1<0)and(abs(rab1)>dopusk)
            then
              begin
                nofvio:=nofvio+1;
                iv[nofvio]:=i;
                if lmax=1 then
                  jv[nofvio]:=2 else jv[nofvio]:=3;

                if abs(rab1)>dis then
                  dis:=abs(rab1);
                end;
              end; {i}
            if kappa>0 then
              begin
                nofvio:=nofvio+1;
                iv[nofvio]:=0;

```

```

jv[nofvio]:=0;
end;

for i:=1 to n do
  begin
    for j:=i+1 to n+8 do
      begin
        rab1:=0;
        if j<=n then rab1:=sqrt(sqr(x[i]-
        x[j])+sqr(y[i]-y[j])+sqr(z[i]-z[j]))-
        (r[i]+r[j])-kappa;

        {rab1:=
        sqrt(sqr(x[i])+sqr(y[i]))+(r[0]-r[i]-
        kappa);}
        { if j=n+1 then rab1:=x[i]-r[i]-
        kappa;
        if j=n+2 then rab1:=y[i]-r[i]-kappa;
        if j=n+3 then rab1:=widy-y[i]-r[i]-
        kappa;
        if j=n+4 then rab1:=widx-x[i]-r[i]-
        kappa;
        if j=n+5 then rab1:=z[i]-r[i]-kappa;
        if (j=n+6)and((tofp=1)or(tofp=3))
        then rab1:=leng-z[i]-r[i]-kappa;
        if (j=n+7)and(tofp=3) then rab1:=
        sqrt(sqr(x[i]-xcyl)+sqr(y[i]-ycyl))+
        (rcyl-r[i])-kappa;}
        // if j=n+8 then rab1:=sqrt(sqr(x[i]-
        xsph)+sqr(y[i]-ysph)+sqr(z[i]-
        zsph))+leng*2-r[i]-kappa;
        if (j=n+8)and(tofp=2) then rab1:=
        sqrt(sqr(x[i]-xsph)+sqr(y[i]-
        ysph)+sqr(z[i]-zsph))+leng-r[i]-kappa;
        if
        (rab1<0)and(abs(rab1)>epsb*1000) dop
        usk {and(ic>0) then
        { if not(mc[i,j]) then}
        begin
          nofvio:=nofvio+1;
          iv[nofvio]:=i;
          jv[nofvio]:=j;
          if abs(rab1)>dis then
            dis:=abs(rab1);
          end;
        end;
        if (r[i]+1e-4<1e-
        3 {+kappa})and(abs(r[i])>eps)and(not(lrh
        [i])) then
          begin
            nofvio:=nofvio+1;
            iv[nofvio]:=1;
            jv[nofvio]:=i;
          end;
          if (r[i]>rin[i]+tol)and(abs(r[i]-
          rin[i])>eps)and(not(lrh[i]))and(not(lprob
          )) then
            begin
              nofvio:=nofvio+1;
              iv[nofvio]:=2;
              jv[nofvio]:=i;
            end;

          end;
        end;
      end;
    end;
  end;
end;

procedure sortm1;
var i,j:integer;
begin
  i1:=0;
  for i:={-2}1 to n do
    for j:=i+1 to n+4 do
      if j>0 then
        if mc[i,j] then
          begin

```

```

i1:=i1+1;
m1[i1]:=i; m2[i1]:=j;
end;
end;

procedure
sortm(mpig,mpjg,mpkg,mpii:integer;md
ofr:double);
label 1;
var i,j:integer;
begin
  if nnm=0 then i:=1 else
    for i:=1 to nnm do
      if (mdofr<mdm[i]) then
        begin
          goto 1;
        end;
      if i=nnm then i:=i+1;
    1;;
    for j:=nnm downto i do
      begin
        mdm[j+1]:=mdm[j];

        igm[j+1]:=igm[j]; jgm[j+1]:=jgm[j]; kgm[
        j+1]:=kgm[j]; iim[j+1]:=iim[j];
        end;
        mdm[i]:=mdofr; igm[i]:=mpig;
        jgm[i]:=mpjg; kgm[i]:=mpkg;
        iim[i]:=mpii;
        nnm:=nnm+1;
        end;
      {block0}
      function dist(i,j:integer):double;
      var d:double;
      begin
        if j=n+1 then d:=x[i]-r[i];
        if j=n+2 then d:=y[i]-r[i];
        if j=n+3 then d:=wid-y[i]-r[i];
        if j=n+4 then d:=leng-x[i]-r[i];
        if j<=n then d:=sqrt(sqr(x[i]-
        x[j])+sqr(y[i]-y[j]))-(r[i]+r[j]);
        dist:=d;
      end;

      var s:double;
      function
      sp(ax,ay,bx,by:double):double; {scalar
      product}
      begin
        sp:=ax*bx+ay*by;
      end;

      procedure to_mps(m,n,nz:integer; var
      a:tiar; var ind1:tiar; var ind2:tiar; var
      b:tm5);
      var i,j,k,ibeg,jbeg,kbeg:integer;
      lj:boolean;
      blj,bhj,cj:real;
      f:text;
      s2:string[8];
      begin
        assign(f,'mps.mps');
        rewrite(f);
        writeln(f,'NAME CONVERTER');
        writeln(f,'ROWS');
        writeln(f,'N COST');
        ibeg:=1000000;
        for i:=1 to m do
          begin
            ibeg:=ibeg+1;
            writeln(f,'G L',ibeg,7);
          end;
          writeln(f,'COLUMNS');
          jbeg:=1000000; kbeg:=1000000;
          for j:=1 to n do

```

```

begin
{if abs(c[j])>1e-9 then}if j=1 then
begin
jbeg:=jbeg+1;
cj:=1;
str(cj:12:10,s2); {delete(s1,12,3);}
{if abs(c[j])>1e-6 then}
writeln(f,' X'j+kbeg:7,' COST
',s2);
end;
lj:=true;
for k:=1 to nz do
if ind2[k]=j then
if abs(a[k])>1e-9 then
begin
lj:=false;
if a[k]>9.999999999e10 then
a[k]:=9.999999999e10;
if a[k]<-9.999999999e10 then a[k]:=
-9.999999999e10;

str(a[k]:12:10,s2); {delete(s1,12,3);}
writeln(f,' X',(ind2[k]+kbeg):7,'
L',(ind1[k]+kbeg):7,' ',s2);
end;
if lj then
writeln(f,' X'j+kbeg:7,' COST
0');
end;
writeln(f,'RHS');
ibeg:=1000000;
for i:=1 to m do
{if abs(b[i])>1e-9 then}
begin
ibeg:=ibeg+1;
if b[i]>9.999999999e10 then
b[i]:=9.999999999e10;
if b[i]<-9.999999999e10 then b[i]:=
-9.999999999e10;

str(b[i]:12:10,s2); {delete(s1,12,3);}
writeln(f,' RHS1 L',ibeg:7,' ',s2);
end;
writeln(f,'BOUNDS');
jbeg:=1000000; blj:=1; bhj:=1;
for j:=1 to n do
begin
jbeg:=jbeg+1;
if j=1 then begin blj:=1000;
bhj:=1000; end else begin blj:=1;
bhi:=1; end;
str(blj:12:10,s2);
writeln(f,' LO BND1 X'jbeg:7,'
',s2);
str(bhj:12:10,s2);
writeln(f,' UP BND1 X'jbeg:7,'
',s2);
end;
writeln(f,'ENDATA');
close(f);
assign(f,'spc');
rewrite(f);
writeln(f,'begin');
{writeln(f,'rows ',m+n+1:12);
writeln(f,'cols ',n+m+n:12);
writeln(f,'elements ',nz+n+n:12);}
writeln(f,'mps file mps.mps');
writeln(f,'log file mps.log');
writeln(f,'solut file mps.res');
writeln(f,'opt tol 1.0D-8');
writeln(f,'presolve 0');
writeln(f,'end');
close(f);
end;

procedure to_bpmpd(m,n,nz:integer; var
a:tiar; var ind1:tiar; var ind2:tiar; var
b:trn5);
var i,j,k,inz,nj:integer;

```

```

blj,bhj,cj:real;
f:text;
begin
assignfile(f,'impdata.txt');
rewrite(f);
big:=1e+30;
inz:=0; qn:=0; qnz:=0;
for j:=1 to n do
begin
obj[j]:=0; ukaz[j]:=1;
nj:=0;
for k:=1 to nz do
if ind2[k]=j then
if abs(a[k])>1e-9 then
begin
nj:=nj+1; inz:=inz+1;
acolidx[inz]:=ind1[k];
acolnzs[inz]:=a[k];
if ((nj=1)and(abs(a[k])-1>1e-
6))or(nj=2) then ukaz[j]:=0;
end;
acolcnt[j]:=nj;
end;
writeln(f,'m');
writeln(f,'m');
writeln(f,'n');
writeln(f,'n');
writeln(f,'nz');
writeln(f,'nz');
writeln(f,'inz');

writeln(f,'qn');
writeln(f,'0');
writeln(f,'qnz');
writeln(f,'0');
writeln(f,'acolcnt');
for j:=1 to n do
writeln(f,'acolcnt[j]');
writeln(f,'acolidx');
for k:=1 to inz do
writeln(f,'acolidx[k]');
writeln(f,'acolnzs');
for k:=1 to inz do
writeln(f,'acolnzs[k]');

for i:=1 to m do
begin
rhs[i]:=b[i];
end;
writeln(f,'rhs');
for i:=1 to m do
writeln(f,'rhsfil');

blj:=1; bhj:=1;
lbound[1]:=blj;ubound[1]:=bhj;
for j:=2 to n do
begin
if {ukaz[j]=1} acolcnt[j]=0 then begin
lbound[j]:=0;ubound[j]:=0; end else
begin
lbound[j]:=blj;ubound[j]:=bhj;
end;
end;

for j:=n+1 to n+m do
begin
lbound[j]:=0;ubound[j]:=big;
end;

if llleng then obj[1]:=1 else obj[1]:=1;
writeln(f,'obj');
for j:=1 to n do
writeln(f,'obj[j]');

writeln(f,'lbound');
for j:=1 to n do
writeln(f,'lbound[j]');
writeln(f,'ubound');
for j:=1 to n do
writeln(f,'ubound[j]');

```

```

writeln(f,'big');
writeln(f,'big');
closefile(f);
end;

procedure
addconstr(i,j:integer;eps:double);
var sqrr:double;
begin
it:=it+1;
or1[it]:=i;or2[it]:=j;
b[it]:=1e-6 {+tolcyl};
{***}
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=1;
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=dimmatrb3;
{888}
if j=n+1 then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=i*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if lr[i] then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
if j=n+2 then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=i*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if lr[i] then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
if j=n+3 then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=i*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if lr[i] then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
if j=n+4 then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=i*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if lr[i] then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i];

```

```

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
if lleng then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=n2+2;

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
{ if j<0 then
begin
end;}
if (j<=n)and(j>0) then
begin
sqrtr:=sqrt(sqr(x[i]-x[j]))+sqr(y[i]-
y[j]));
if sqrtr>eps then begin
nz:=nz+1;
a[nz]:=(x[i]-x[j])/sqrtr;
ind1[nz]:=it; ind2[nz]:=i*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
nz:=nz+1;
a[nz]:=(x[i]-x[j])/sqrtr;
ind1[nz]:=it; ind2[nz]:=j*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
nz:=nz+1;
a[nz]:=(y[i]-y[j])/sqrtr;
ind1[nz]:=it; ind2[nz]:=i*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
nz:=nz+1;
a[nz]:=(y[i]-y[j])/sqrtr;
ind1[nz]:=it; ind2[nz]:=j*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
if (lr[i]) then
begin
nz:=nz+1;
a[nz]:=1;
ind1[nz]:=it; ind2[nz]:=nofvr[i];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
if (lr[j]) then
begin
nz:=nz+1;
a[nz]:=1;
ind1[nz]:=it; ind2[nz]:=nofvr[j];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;

function nor:integer;
var i,ii:integer;
begin
ii:=0;
for i:=1 to n do
if lr[i] then ii:=ii+1;
nor:=ii;
end;

procedure rvar;
var i,il:integer;
begin
il:=0;
for i:=1 to n do
begin
if lr[i] then
begin
il:=il+1;
nofvr[i]:=i+1+n2+1;
end else nofvr[i]:=0;
end;
dimmatrb3:=i1+1+n2+1;

```

```

end;

procedure rvar1;
var i,il:integer;
begin
il:=0;
for i:=1 to n do
begin
if lr[i] then
begin
il:=il+1;
nofvr[i]:=i1+n2+1;
end else nofvr[i]:=0;
end;
{if lleng then i1:=i1+1;}
dimmatrb3:=i1+1+n2;
end;

procedure fomcoef(eps1:double);
var i,j:integer;
begin
it:=0; nz:=0;
for i:=1 to n2+n+1 do hc[i]:=0;

for i:=1 to n do
for j:=i+1 to n+4 do
if mc[i,j] then
begin
addconstr(i,j,eps1);
end;

for i:=1 to n do
if lr[i] then
if lrr[i] then
begin
it:=it+1; nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i]; b[it]:=0;
end;
for i:=1 to n do
if lr[i] then
if lrr[i] then
begin
it:=it+1; nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i]; b[it]:=0;

end;

begin
it:=it+1;
nz:=nz+1; b[it]:=0;
a[nz]:=1; ind1[nz]:=it; ind2[nz]:=1;
for i:=1 to dimmatrb3-1 do
if (abs(xtargin[i])>eps) then
begin
nz:=nz+1;
a[nz]:=xtargin[i]; ind1[nz]:=it;
ind2[nz]:=i+1;
end;
end;

to_mps(it,dimmatrb3,nz,a,ind1,ind2,b);
end;

procedure
addconstr1(i,j:integer;eps:double);
begin
it:=it+1;
or1[it]:=i; or2[it]:=j;
b[it]:=0; {1e-5} if lprob then
b[it]:=1e-2;

if j=n+1 then
begin

```

```

nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=i*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if lr[i] then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
if j=n+2 then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=i*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if lr[i] then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
if j=n+3 then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=i*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if lr[i] then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
if j=n+4 then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=i*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if lr[i] then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=nofvr[i];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
if lleng then
begin
nz:=nz+1;
a[nz]:=1; ind1[nz]:=it;
ind2[nz]:=n2+2+1;

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;
if (j<=n)and(j>0) then
begin
if not(lleng)and((lr[i])or(lr[j])) then
begin
nz:=nz+1;
a[nz]:=-0.1; ind1[nz]:=it; ind2[nz]:=1;
end;

nz:=nz+1;

```

```

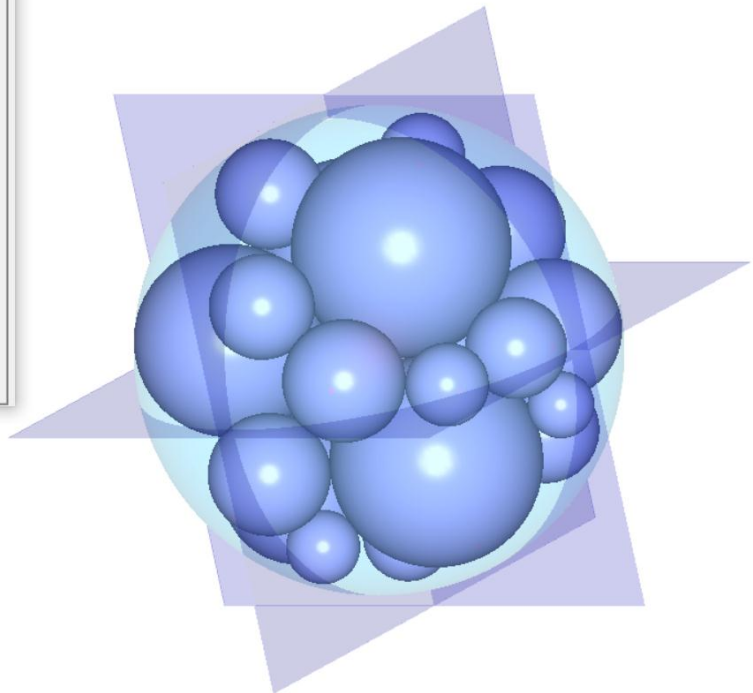
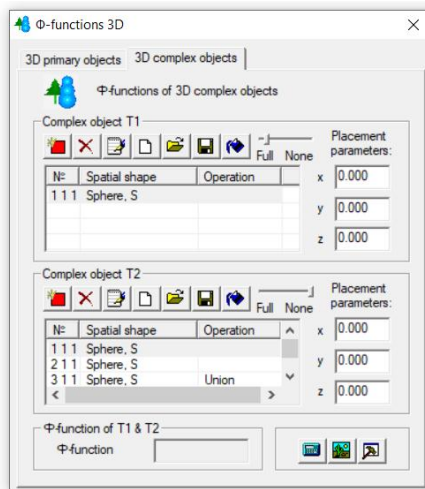
a[nz]:=2*(x[i]-x[j]); ind1[nz]:=it;
ind2[nz]:=i*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
nz:=nz+1;
a[nz]:=2*(x[i]-x[j]); ind1[nz]:=it;
ind2[nz]:=j*2;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
nz:=nz+1;
a[nz]:=2*(y[i]-y[j]); ind1[nz]:=it;
ind2[nz]:=i*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
nz:=nz+1;
a[nz]:=2*(y[i]-y[j]); ind1[nz]:=it;
ind2[nz]:=j*2+1;
hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
if (lr[i]) then
begin
nz:=nz+1;
a[nz]:=2*(r[i]+r[j]);
ind1[nz]:=it; ind2[nz]:=nofvr[i];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
if (lr[j]) then
begin
nz:=nz+1;
a[nz]:=2*(r[i]+r[j]);
ind1[nz]:=it; ind2[nz]:=nofvr[j];

hc[ind2[nz]]:=hc[ind2[nz]]+a[nz];
end;
end;

if
((lhc)and(abs(hc[i])>eps))or((not(lhc))an
d(abs(xtargin[i])>eps)) then
begin
nz:=nz+1;
if lhc then
a[nz]:=hc[i] else
a[nz]:=xtargin[i]; ind1[nz]:=it;
ind2[nz]:=i+1;
end;
end;
if not(lleng) then
begin
nz:=nz+1;
a[nz]:=-1; ind1[nz]:=it; ind2[nz]:=1;
end;
if {not(lleng)}lprob then
if mc[-1,-1] then
begin
it:=it+1; b[it]:=0;
for i:=1 to n do
if lr[i] then
begin
nz:=nz+1; a[nz]:=-1; ind1[nz]:=it;
ind2[nz]:=nofvr[i];
end;
end;
{if lprob then
begin

```

Рисунок Б.2 – Программа Φ -function 3D

ВІДОМІСТЬ АТЕСТАЦІЙНОЇ РОБОТИ

Позначення	Найменування	Дод. відомості
	Текстові документи	
1	Пояснювальна записка	76 с.
2	Презентаційний матеріал	25 с.
	Інші документи	
3	Роздруківки програм	7 с.
4	Рецензія	2 с.
5	Відгук керівника	1 с.

					Упакування нерівних гіперкуль у гіперсферичному контейнері						
Змін	Арк.	Номер докум.	Підп.	Дата	(Тема роботи) Відомість атестаційної роботи				Аркуш	Аркушів	
Розроб.		Луцьова О.С.									
Перевір.		Стоян Ю.Г.									
Н. контр.		Сидоров М.В.				ХНУРЕ Кафедра ПМ					
Затв.		Гевяшев А.Д.									