V. Malakhovsky

SUBSETS OF PRIME NUMBERES IN GENERALIZED ARITHMETICAL PROGRESSION OF HIGH ORDER

Sequenses $\left(a_n^{(k)}\right)$ of numbers defined by recurrent relation $a_{n+1}^{(k)} = a_n^{(k)} + d \cdot n^k$, where $a_1^{(k)} = p$ is an odd prime number and d is an even positive number are considered ($k \leq 20$, $p < 10^4$, $d \leq 200$). It is shown that for even k the number d for each subset $\left\{a_1^{(k)}, a_2^{(k)}, \ldots, a_m^{(k)}\right\}$ of prime numbers where $m \geq 5$ is divisible by 6.

УДК 514.75

Н.В. Малаховский

(Российский государственный университет им. И. Канта, г. Калининград)

КОМПЬЮТЕРНЫЕ ПРОГРАММЫ ИССЛЕДОВАНИЯ МЕТОДОМ КАРТАНА МНОГООБРАЗИЙ ФИГУР В ОДНОРОДНЫХ ПРОСТРАНСТВАХ

Разработаны компьютерные программы исследования систем пфаффовых уравнений в однородных и обобщенных пространствах, позволяющие установить совместность или несовместность исходной системы, определить (в случае совместности) произвол ее решения. Рассмотрены конкретные примеры применения данных в работе компьютерных программ к исследованию поверхностей в евклидовом и проективном пространствах и конгруэнций эллипсоидов в аффинном пространстве.

Интенсивное развитие компьютерной техники и совершенствование методов программирования способствовали внедрению компьютерных методов вычисления в классические направления теоретической математики. Использование пакета программ Maple V Release 4.00a позволило составить компьютерные программы исследования методом Картана систем пфаффовых уравнений, задающих многообразия фигур в однородных и обобщенных пространствах. В данной статье представлена компьютерная программа исследования таких систем пфаффовых уравнений в евклидовом, аффинном и проективном пространствах произвольной конечной размерности.

Для использования программы необходимо задать систему уравнений Пфаффа исследуемого многообразия в каноническом репере, обозначив неизвестные функции символами $z_1, z_2, z_3, \ldots$, а константы символами $\alpha_1, \alpha_2, \alpha_3, \ldots$ Кроме того, определить какие из форм Пфаффа или их линейных комбинаций являются базисными, и обозначить их символами $\theta_1, \theta_2, \theta_3, \ldots$, а также указать определяемые задачей связи на неизвестные функции (если они имеются), тип пространства, его размерность, число неизвестных функций и число констант.

Компьютерная программа состоит из вводной части и основной программы. Вводные части для евклидова, проективного и аффинного пространств отличаются друг от друга. Для исследования дифференцируемого многообразия необходимо в соответствующей вводной части программы задать начальные данные, указанные выше, после этого запустить основную программу. Так, например, вводная часть компьютерной программы исследования системы дифференциальных уравнений минимальной линейчатой поверхности (прямого геликоида) в евклидовом пространстве

$$\omega_1^3 - z_1\omega^2 = 0,\ \ \omega_2^3 - z_1\omega^1 = 0,\ \omega_1^2 - z_2\omega^2 = 0,\ \ \omega^3 = 0,$$
$$\omega_1^2 + \omega_2^1 = 0,\ \ \omega_1^3 + \omega_3^1 = 0,\ \omega_2^3 + \omega_3^2 = 0,\ \ \omega_1^1 = 0,\ \ \omega_2^2 = 0,\ \ \omega_3^3 = 0$$

имеет вид

```
> Equations:={omega[[1],[3]]-z[1]*omega[[],[2]]=0, omega[[2],[3]]-
z[1]*omega[[],[1]]=0, ome-
ga[[1],[2]]-z[2]*omega[[],[2]]=0,omega[[],[3]]=0, ome-
```

```
ga[[1],[2]]+omega[[2],[1]]=0, omega[[1],[3]]+omega[[3],[1]] =0, ome-
ga[[2],[3]]+omega[[3],[2]]
=0,omega[[1],[1]]=0,omega[[2],[2]]=0,omega[[3],[3]]=0};
> Basis:={theta[1]=omega[[],[1]],theta[2]=omega[[],[2]]};
> Conditions:={z[1]<>0};
> Space:=euclid; Dimension:=3;
> Functions:=2;Constants:=0;
```

Вводная часть программы исследования системы дифференциальных уравнений подкласса проективных сфер в проективном пространстве

$$\omega_0^0 - \frac{1}{2}\left(z_1\omega^1 + z_2\omega^2\right) = 0,\ \omega_1^1 - \frac{1}{6}\left(z_2\omega^2 - z_1\omega^1\right) = 0,\ \omega_2^2 + \omega_1^1 = 0,$$

$$\omega_0^0 + \omega_3^3 = 0,\ \omega_2^3 - \omega^1 = 0,\ \ \omega_1^0 + \frac{z_3}{z_4}\omega^1 = 0,\ \ \omega_2^0 + z_3 z_4\omega^2 = 0,$$

$$\omega_1^2 - \omega^1 = 0,\ \ \omega_2^1 - \omega^2 = 0,\ \ \omega_1^3 - \omega^2 = 0,\ \omega_3^1 - \omega_2^0 = 0,\ \ \omega_3^2 - \omega_1^0 = 0,$$

$$\omega_3^0 + z_3 z_4\omega^1 + \frac{z_3}{z_4}\omega^2 = 0,\ \ \omega_0^3 = 0,\ \ d(z_3) - \omega_3^0 - 2z_3\omega_3^3 = 0,$$

$$d(z_4) + \omega^2 - 2z_4\omega_2^2 - z_4^2\omega^1 = 0$$

имеет вид

```
> Equations:={omega[[0],[0]] - 1/2*(z[1]*omega[[],[1]]+
z[2]*omega[[],[2]])=0,omega[[1],[1]] - 1/6*(z[2]* ome-
ga[[],[2]]-z[1]*omega[[],[1]])=0,omega[[2],[2]]+omega[[1],[1]]=0,o-
mega[[3],[3]]+omega[[0],[0]]=0,
omega[[1],[0]]+z[3]/z[4]*omega[[],[1]]=0,omega[[2],[0]]+z[3]*z[4]*ome
ga[[],[2]]=0, omega[[1],[2]]- omega[[],[1]]=0,
omega[[2],[1]]-omega[[],[2]]=0,omega[[1],[3]]-omega[[],[2]]=0,
omega[[2],[3]]-omega[[],[1]]=0,
omega[[3],[1]]-omega[[2],[0]]=0,omega[[3],[2]]-omega[[1],[0]]=0, ome-
ga[[3],[0]]+z[3]*z[4]*omega[[],[1]]+ z[3]/z[4]*omega[[],[2]]=0,
omega[[0],[3]]=0, d(z[3])-omega[[3],[0]]-2*z[3]*omega[[3],[3]]=0,
d(z[4])+ omega[[],[2]]-2*z[4]*omega[[2],[2]]-z[4]^2*omega[[],[1]]=0};
> Basis:={theta[1]=omega[[],[1]],theta[2]=omega[[],[2]]};
> Conditions:={};
> Space:=projective; Dimension:=3;
> Functions:=4;Constants:=0;
```

Вводная часть программы исследования дифференциальных уравнений подкласса конгруэнций эллипсоидов в аффинном пространстве

$$\omega_1^3 + (z_1+1)\omega^1 = 0, \quad \omega^3 = 0, \quad \omega_2^1 - z_3\omega^1 - z_5\omega^2 = 0,$$
$$\omega_1^2 - z_4\omega^1 - z_3\omega^2 = 0, \quad \omega_3^2 - z_2\omega^2 = 0, \quad \omega_3^1 - z_1\omega^1 = 0,$$
$$\omega_2^3 + (z_2+1)\omega^2 = 0, \quad \omega_2^2 + \omega^2 = 0, \quad \omega_1^1 + \omega^1 = 0, \quad \omega_3^3 = 0$$

имеет вид

```
> Equations:={omega[[1],[3]] +(z[1]+1)*omega[[],[1]]=0, omega[[],[3]] =
0, omega[[2],[1]] - z[3]*omega[[],[1]]-z[5]*omega[[],[2]]=0,
omega[[1],[2]] -z[4]*omega[[],[1]]-z[3]*omega[[],[2]]=0, omega[[3],[2]]
-z[2]*omega[[],[2]]=0, omega[[3],[1]] - z[1]*omega[[],[1]]=0,
omega[[2],[3]] +(z[2]+1)*omega[[],[2]]=0, omega[[2],[2]]
+omega[[],[2]]=0, omega[[1],[1]] +omega[[],[1]]=0, omega[[3],[3]] = 0};
> Basis:={theta[1]=omega[[],[1]],theta[2]=omega[[],[2]]};
> Conditions:={z[1]<>z[2]};
> Space:=affine; Dimension:=3;
> Functions:=5;Constants:=0;
```

Основная же программа имеет следующий вид:

```
> with(difforms):with(linalg):with(combinat): def-
form(omega=1,theta=1,Z=1,z=0,w=0,
> seq(alpha[j]=const,j=1..Constants)): m:=nops(Basis):f:=Functions:
dim:=Dimension:
> if Space=euclid then
>   for i while i<=dim
>   do d(omega[[],[i]]):=sum('omega[[],[k]]&^omega[[k],[i]]','k'=1..dim):
>      for j while j<=dim
>      do
d(omega[[i],[j]]):=sum('omega[[i],[k]]&^omega[[k],[j]]','k'=1..dim)
>      od:
>   od:solve({op(Basis),op(Equations)},
>
{seq(op([seq(omega[[i],[j]],i=1..dim),omega[[],[j]]]),j=1..dim),seq(d(z[j]),
j=1..f)}):
```

```
>    remove(has,'',{seq(d(z[j])=d(z[j]),j=1..f)}):
> fi:
> if Space=affine then
>    for i while i<= dim
>    do d(omega[[],[i]]):=sum('omega[[],[k]]&^omega[[k],[i]]','k'=1..dim):
>       for j while j<= dim
>       do
d(omega[[i],[j]]):=sum('omega[[i],[k]]&^omega[[k],[j]]','k'=1..dim):
>       od:
>    od:
>    solve({op(Basis),op(Equations)},
>    {seq(op([seq(omega[[i],[j]],i=1..
dim),omega[[],[j]]]),j=1..dim),seq(d(z[j]),j=1..f)}):
>    remove(has,'',{seq(d(z[j])=d(z[j]),j=1..f)}):
> fi:
> if Space=projective then
>    for i from 0 to dim
>    do for j from 0 to dim
>       do
dom[[i],[j]]:=d(omega[[i],[j]])=sum('omega[[i],[k]]&^omega[[k],[j]]','k'=
0..dim):
>       od:
>    od:
>    (seq(omega[[0],[j]]=omega[[],[j]],j=1..m)):assign(''):
>    (seq(seq(dom[[i],[j]],i=0.. dim),j=0..dim)):assign(''):
>    solve({op(Basis),op(Equations)},{seq(seq(omega[[i],[j]],
>    i=0..dim),j=0..dim),seq(d(z[j]),j=1..f)}):
>    remove(has,'',{seq(d(z[j])=d(z[j]),j=1..f)}):
> fi:
> eqns[1][1]:=expand(''):solve(Basis,{seq(theta[j],j=1..m)}):
> seq(simplify(d(op(l,''))),l=1..m):assign(''):
> fun[1][1]:=f:Sv[1][1]:={{}}:P:=1:G:=1:
> for p while p<=P
> do g1:=0:for g while g<=G
>    do for h while h<=nops(Sv[p][g])
>       do f:=fun[p][g]:
>          {seq('d'(select(has,lhs(op(j,op(h,Sv[p][g]))),
>          {seq(z[k],k=1..f)})),j=1..nops(op(h,Sv[p][g])))}:
>          S[1]:=simplify(subs(op(h,Sv[p][g]), remove(has,eqns[p][g],''))):
>          s:=nops(S[1])-m: subs(op(h,Sv[p][g]),eqns[p][g]):
```

```
>        {seq(d(lhs(op(j,''))-rhs(op(j,''))),j=1..nops(''))}:
>        dS[1]:=simplify(subs(op(h,Sv[p][g]),'')):assign(S[1]):K:=0:J:=1:
>        for i while i<=J
>        do
dS[i]:=remove(has,simpform(dS[i]),0):ddS[i]:=dS[i]:K:=K+1:b:=1:
>          for a while a<=f
>         do if op(select(has,dS[i],d(z[a])))<>NULL then  c[b]:=a:b:=b+1 fi:
>          od:q:=b-1:seqf[i]:=[seq(c[j],j=1..b-1)]:
>          if op(seqf[i])<>NULL then k:=1:rnk[0]:=0:
>            for l while l<=m-1
>            do Sl:=subs({seq(theta[i]=theta[i[l]],i=1..m),
seq(d(z[k])=Z[k],k=seqf[i])},dS[i]):
>               dSl:=simpform(d(Sl)):
>              for b while b<=nops(dSl)
>              do for c in seqf[i]
>                do simpform(subs(d(Z[c])=1,op(b,dSl))):
>                  n[k]:=simpform(subs(`&^`=0,'')):k:=k+1:
>                od:
>              od:rnk[l]:=rank(matrix(nops(dSl)*l,q,
[seq(n[k],k=1..q*nops(dSl)*l)])):
>              sk[l]:=rnk[l]-rnk[l-1]:
>            od:sk[l]:=q-sum('sk[k]','k'=1..l-1):
Q:=sum('sk[k]*k','k'=1..m):t:=f+1:
>            for j in seqf[i]
>            do d(z[j]):=sum('z[k]*theta[k-t+1]','k'=t..t+m-1):t:=t+m:
>            od:dS[i]:=remove(has,simpform(dS[i]),0):
>            t:=f+q*m:per:=permute(m,2):k:=1:typ:=0:
>            for a while a<=nops(dS[i])
>            do for b while b<=nops(per)
>              do e[k]:=coeff(op(a,dS[i]),'`&^`'(theta[op(1,op(b,per))], the-
ta[op(2,op(b,per))])):
>                if e[k]<>0 and type(e[k],constant)=true then typ:=typ+1 fi:
k:=k+1
>              od:
>            od:
>            if typ<>0 then seq(unassign('d'(z[j])),j=1..f):
>              unassign('omega'): print(`при выполнении равенств`):
>              print(op(op(h,Sv[p][g]))): print(`система уравнений Пфаффа`):
>              print(subs({op(op(h,Sv[p][g])),op(Basis)},eqns[p][g])):
>              print(`противоречива.`): break
```

```
>            fi:
>            sq:=expand(remove(has,{seq(e[i],i=1..k-1)},0)):nsq:=nops(sq):
>            A:=matrix(nsq,q*m,[seq(seq(coeff(op(k,sq),z[j]),
j=f+1..t),k=1..nsq)]):
>            V:=vector(nsq,subs([]=0,[seq(-remove(has,op(k,sq)+[],
>            {seq(z[j],j=f+1..t)}),k=1..nsq)])):
>            B:=concat(A,V):rA:=rank(A):rB:=rank(B):
>            if rA<rB then g1:=g1+1:p1:=p+1:
>              rw:=[seq(j,j=1..nsq)]:cl:=[seq(j,j=1..m*q)]:k:=1:
>              for a in rw
>              do for b in cl
>                do aa[k]:=a:bb[k]:=b:
>                  sbm:=submatrix(A,[seq(aa[j],j=1..k)],
[seq(bb[j],j=1..k)]):dt:=det(sbm):
>                 if dt<>0 then  rw:=remove(has,rw,a):cl:=remove(has,cl,b):
k:=k+1:break: fi:
>                od:
>              od:
>              remove(has,[seq(j,j=1..nsq)],[seq(aa[j],j=1..k-1)]):
>              seq(det(submatrix(B,[seq(aa[j],j=1..k-1),l],
>              [seq(bb[j],j=1..k-1),m*q+1])),l="):remove(has,{"},0):
>
{seq(solve({op(k,"),op(op(h,Sv[p][g]))},{seq(z[j],j=1..f)}),k=1..nops("))}
:
>             seq(remove(has,op(j,"),{seq(z[k]=z[k],k=1..f)}), j=1..nops(")):
>              sel:=select(has,{"},z):c:=0:
>              for b while b<=nops(sel)
>              do sub:=subs(op(b,sel),Conditions):
>                {seq(simplify(lhs(op(j,sub))-rhs(op(j,sub))),
>                j=1..nops(sub))}:
>                if op(select(has,",0))<>0 then c:=c+1:Svn[c]:=op(b,sel) fi:
>              od:Sv[p1][g1]:={seq(Svn[j],j=1..c)}: unassign('omega'):
>              seq(unassign('d'(z[j])),j=1..f):
eqns[p1][g1]:={seq(op(S[j]),j=1..i)}:
>              fun[p1][g1]:=f:break
>            fi:
>            if rA=rB then slv:={solve(sq,{seq(z[k],k=f+1..t)})}:
>              if nops(slv)>1 then p1:=p+1:g1:=g1+1:
>               seq(unassign('d'(z[j])), j=remove(has,[seq(j,j=1..f)],seqf[i])):
>                eqns[p1][g1]:={seq(op(S[j]),j=1..i),
seq('d'(z[j])=d(z[j]),j=seqf[i])}:
```

```
>                  seq(unassign('d'(z[j])),j=seqf[i]):
{seq(op(k,slv),k=1..nops(slv))}:
>                  Sv[p1][g1]:={seq(remove(has,op(j,''),
{seq(z[k]=z[k],k=1..t)}),j=1..nops(''))}:
>                  fun[p1][g1]:=t:break
>                fi:Ln:=nops(remove(has,op(slv), {seq(z[k]=z[k],k=f+1..t)})):
>                N:=m*q-Ln:assign(slv)
>              fi:seqs:=seqf[i]:subs(z=w,[seq(z[j],j=f+1..t)]):
>              sct:=select(has,'',[seq(w[j],j=f+1..t)]): k:=nops(sct):
>              for a while a<=k
>              do
>                z[op(1,op(a,sct))]:=w[f+a]
>              od:
>              S[i+1]:={seq(d(w[j])=simpform(d(z[j])),j=seqs)}:
>              dS[i+1]:={seq(simpform(d(lhs(op(j,S[i+1]))-
>              rhs(op(j,S[i+1])))),j=1..nops(S[i+1]))}:
seq(unassign('d'(z[j])),j=1..f):
>              unassign('z'):defform(z=0): S[i+1]:=subs(w=z,S[i+1]):
>              dS[i+1]:=subs(w=z,dS[i+1]):f:=f+k:
>              if i>1 then
>                seq(type(select(has,{op(j,S[i])},
[seq(z[j],j=[seq(op(seqf[j]),j=2..i)])]),type),
>                j=1..nops(S[i])): select(has,[''],true):s:=nops(''):
>              fi:
>            else seq(unassign('d'(z[j])),j=1..f):unassign('omega'):
>              if op(op(h,Sv[p][g]))<>NULL then print(`при выполнении
равенств`):
>                print(op(op(h,Sv[p][g]))):
>              fi:print(`система уравнений Пфаффа`):
>                print(subs({op(op(h,Sv[p][g])),op(Basis)},eqns[p][g])):
>              if op(ddS[i])=NULL then print(`вполне интегрируема`):
>                break else print(`противоречива.`): break
>              fi:
>            fi:
>            if Q=N then seq(unassign('d'(z[j])),j=1..f): unassign('Omega'):
>              if i>1 then seq({print(j-`ое продолжение`),  print(S[j])},j=1..i)
fi:
>              if op(op(h,Sv[p][g]))<>NULL then print(`при выполнении
равенств`):
>                print(op(op(h,Sv[p][g])))
```

```
>         fi:
>         print(`система уравнений Пфаффа`):
>         print(subs({op(op(h,Sv[p][g])),op(Basis)},eqns[p][g])): print(`в инволюции`):
>         print(`и задаёт интегральное многообразие с произволом `):
>         seq(print(sprintf(`%d  функций   %g  аргументов`,sk[k],k)),k=[seq(-j,j=-m..-1)]):
>         print(sprintf(`и  %f  произвольных констант`,s)):
>         print(`итоговое замыкание имеет вид`):
>         print(simpform(subs(Basis,{seq(op(j,ddS[i])=0,
j=1..nops(ddS[i]))}))):break
>       fi:if K=i then J:=K+1 fi:
>     od:seq(unassign('d'(z[j])),j=1..f):unassign('omega'):
>   od:
>  od:G:=g1:if g1=0 then break else P:=P+1 fi:
> od:
```

Список литературы

1. *Малаховский В.С.* Введение в теорию внешних форм. Калининград, 1978.
2. *Малаховский В.С.* Введение в теорию внешних форм. Ч. 2. Калининград, 1980.
3. *Юрова Е.П.* Конгруенции эллипсоидов с кратными фокальными поверхностями // Диф. геом. многообр. фигур. Калининград, 2004. Вып. 35. С. 162—169.

N. Malakhovsky

COMPUTER PROGRAMS OF RESEARCH BY CARTAN'S METHOD MANIFOLDS OF FIGURES IN HOMOGENEOUS SPACES

Computer programs of research of Pfaffian's systems of equations in homogeneous and generalized spaces are given, allowing to establish compatibility or noncompatibility of the given system and to define in a case of compatibility an arbitrariness of its solution. Concrete examples of application of these computer programs to research of surfaces in euclidian and projective spaces and congruences of ellipsoids in affine space are considered.