Приложение.

Реализация метода каскадного интегрирования Лапласа для строго гиперболических систем в пакете Maple 18.

```
> restart:
with(LinearAlgebra):

#Вводим число вычисляемых инвариантов Лапласа
N:=0:

#Ввод функций гиперболической системы
#(если требуется решить произвольную
#строго гиперболическую систему, то данные следует вводить здесь)
printf("Вид матрицы системы"):
a[1,1]:=simplify(diff(AA,u)):a[1,2]:=simplify(diff(AA,v)):
a[2,1]:=simplify(diff(BB,u)):a[2,2]:=simplify(diff(BB,v)):
M:=Matrix([[a[1,1],a[1,2]],[a[2,1],a[2,2]]]);

#Коэффициенты уравнения 2го порядка
printf("Коэффициенты уравнения второго порядка в плоскости годографа"):
A:=simplify(a[1,2],symbolic);
B:=simplify(a[2,2]-a[1,1],symbolic);
C:=-simplify(a[2,1],symbolic);
DD:=simplify(diff(a[1,2],u)+diff(a[2,2],v),symbolic);
E:=-(simplify(diff(a[2,1],v)+diff(a[1,1],u),symbolic));

#Решение характеристических уравнений, замена переменных
printf("Замена переменных"):
diff(v(u), u) = subs(v = v(u), factor(simplify((B+simplify(factor(expand(-
4*A*C+B^2))^(1/2), symbolic))/(2*A), symbolic)));dsolve(%, v(u)):
etaa:=sort(expand(simplify(subs(v(u)=v,solve(%, _C1))))):print(eta=%):
diff(v(u), u) = subs(v = v(u), factor(simplify((B-simplify(factor(expand(-
4*A*C+B^2))^(1/2), symbolic))/(2*A), symbolic)));dsolve(%, v(u)):
xii:=sort(expand(simplify(subs(v(u)=v,solve(%, _C1))))):print(xi=%):

#Якобиан преобразования
printf("Якобиан преобразования"):
Determinant(Matrix([[diff(xii,u),diff(xii,v)],[diff(etaa,u),diff(etaa,v)]]));
SSS2:=simplify(solve({xii=xii1,etaa=etaa1},{u,v}), symbolic):

if SSS2=NULL then
printf("Программе Maple не удалось выразить старые переменные через новые
\n"):
printf("Введите выражение старых переменных через новые под комментарием
Ввести вручную2 вместо u1,v1 \n"):
printf("через переменные xii,etaa \n"):
#Ввести вручную
SSS2:={u=u1,v=v1}:
else
printf("Выражение старых переменных через новые"):
print(subs(xii1=xi,etaa1=eta,SSS2[1])):
print(subs(xii1=xi,etaa1=eta,SSS2[2])):
end if:

#Коэффициенты канонического уравнения в старых переменных
printf("Коэффициенты канонического уравнения в старых переменных"):
2*A*diff(xii,u)*diff(etaa,u)+B*(diff(xii,u)*diff(etaa,v)+diff(xii,v)*diff(eta
a,u))+2*C*diff(xii,v)*diff(etaa,v): B_:=simplify(%,symbolic):
A*diff(xii,u,u)+B*diff(xii,u,v)+C*diff(xii,v,v)+DD*diff(xii,u)+E*diff(xii,v):
D_:=simplify(%,symbolic):print('a_'=D_/B_):
```

```
A*diff(etaa,u,u)+B*diff(etaa,u,v)+C*diff(etaa,v,v)+DD*diff(etaa,u)+E*diff(etaa,v): E_:=simplify(%,symbolic):print('b_'=E_/B_):

#Коэффициенты канонического уравнения в новых переменных
printf("Коэффициенты канонического уравнения в новых переменных"):

a_:=simplify(subs(subs(xii1=xi,etaa1=eta,SSS2[1]),subs(xii1=xi,etaa1=eta,SSS2[2]),D_/B_),symbolic);
b_:=simplify(subs(subs(xii1=xi,etaa1=eta,SSS2[1]),subs(xii1=xi,etaa1=eta,SSS2[2]),E_/B_),symbolic);

k:=0:m:=0:j:=N+1:
h[0]:=simplify(diff(a_,xi)+a_*b_,symbolic):
h[-1]:=simplify(diff(b_,eta)+a_*b_,symbolic):

if h[0]=0 and h[-1]=0 then
m:=1:j:=0:
printf("Оба инварианта Лапласа в уравнение E[%d] равны нулю",0):

elif h[-1]=0 then
printf("Левый инвариант Лапласа в уравнение E[%d] равен нулю",0):
j:=0:m:=2:

elif h[0]=0 then
printf("Правый инвариант Лапласа в уравнение E[%d] равен нулю",0):
j:=0:m:=3:
else

for i from 0 by 1 to N do
h[i+1]:=2*h[i]-h[i-1]-diff(ln(h[i]),xi,eta):

if h[i]=0 and h[i+1]=0 then
m:=1:j:=i:
printf("Оба инварианта Лапласа в уравнение E[%d] равны нулю",i):

elif h[i]=0 then
printf("Левый инвариант Лапласа в уравнение E[%d] равен нулю",i):
j:=i:m:=2:
elif h[i+1]=0 then
printf("Правый инвариант Лапласа в уравнение E[%d] равен нулю",i):
j:=i:m:=3:
end if:

end do:
end if:

if j=N+1 then
printf("Введенное число, рассчитываемых членов ряда Лапласа не достаточно для нахождения решения"):
else
if j>=0 then
if m=1 then
t[j]:=simplify(subs(tteta=eta,txi=xi,simplify((exp(-(int(a_,eta)+int(b_,xi)))*(X_(txi)+Y_(tteta))))),symbolic):
for i from j by -1 to 1 do
t[i]:=(diff(t[i-1],xi)+b_*t[i-1])/h[i]:
end do:
end if:

if m=2 then
t[j]:=simplify(subs(tteta=eta,txi=xi,subs(eta=etaa,xi=xii,simplify((exp(-int(a_,eta)))*(X_(txi)+int(Y_(tteta)*exp(int(a_,eta)-int(b_,xi)),tteta))))),symbolic):
for i from j by -1 to 1 do
```

```
t[i]:=(diff(t[i-1],xi)+b_*t[i-1])/h[i]:
end do:
end if:

if m=3 then
t[j]:=simplify(subs(tteta=eta,txi=xi,simplify((exp(-
int(b_,xi)))*(Y_(tteta)+int(X_(txi)*exp(-(int(a_,eta)-
int(b_,xi))),txi)))),symbolic):
for i from j by -1 to 1 do
t[i]:=(diff(t[i-1],xi)+b_*t[i-1])/h[i]:
end do:
end if:
end if:

if j<0 then
if m=1 then
t[j]:=simplify(subs(tteta=eta,txi=xi,simplify((exp(-
(int(a_,eta)+int(b_,xi)))*(X_(txi)+Y_(tteta))))),symbolic):
for i from 1 by 1 to j do
t[i]:=(diff(t[i-1],eta)+a_*t[i-1])/h[-i-1]:
end do:
end if:

if m=2 then
t[j]:=simplify(subs(tteta=eta,txi=xi,subs(eta=etaa,xi=xii,simplify((exp(-
int(a_,eta)))*(X_(txi)+int(Y_(tteta)*exp(int(a_,eta)-
int(b_,xi)),tteta))))),symbolic):
for i from 1 by 1 to j do
t[i]:=(diff(t[i-1],eta)+a_*t[i-1])/h[-i-1]:
end do:
end if:

if m=3 then
t[j]:=simplify(subs(tteta=eta,txi=xi,simplify((exp(-
int(b_,xi)))*(Y_(tteta)+int(X_(txi)*exp(-(int(a_,eta)-
int(b_,xi))),txi)))),symbolic):
for i from 1 by 1 to j do
t[i]:=(diff(t[i-1],eta)+a_*t[i-1])/h[-i-1]:
end do:
end if:
end if:

XV:=subs(subs(xii1=xi,etaa1=eta,SSS2[1]),subs(xii1=xi,etaa1=eta,SSS2[2]),diff
(x(xi,eta),xi)*diff(xii,v)+diff(x(xi,eta),eta)*diff(etaa,v)):
XU:=subs(subs(xii1=xi,etaa1=eta,SSS2[1]),subs(xii1=xi,etaa1=eta,SSS2[2]),diff
(x(xi,eta),xi)*diff(xii,u)+diff(x(xi,eta),eta)*diff(etaa,u)):
TV:=subs(subs(xii1=xi,etaa1=eta,SSS2[1]),subs(xii1=xi,etaa1=eta,SSS2[2]),diff
(t[0],xi)*diff(xii,v)+diff(t[0],eta)*diff(etaa,v)):
TU:=subs(subs(xii1=xi,etaa1=eta,SSS2[1]),subs(xii1=xi,etaa1=eta,SSS2[2]),diff
(t[0],xi)*diff(xii,u)+diff(t[0],eta)*diff(etaa,u)):

for i from 1 by 1 to 2 do
for j from 1 by 1 to 2 do
AA[i,j]:=subs(subs(xii1=xi,etaa1=eta,SSS2[1]),subs(xii1=xi,etaa1=eta,SSS2[2])
,a[i,j]):
end do:
end do:

sys:={simplify(XV-AA[1,1]*TV+AA[1,2]*TU=0,symbolic),
simplify(XU+AA[2,1]*TV-AA[2,2]*TU=0,symbolic)}:
end if:
```