Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, No. 2, September : 2023

PROGRAM IN SAGEMATH TO REDUCE ORDER OF SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS WITH KNOWN SYMMETRIES

Vishwas Khare Assistant Professor, *Department of Mathematics* SSR College of ACS, Silvassavskssr@gmail.com

Abstract
In computer algebra system open-source software Sage-Math, a programme is built to use the symmetry approach for differential equations to lower order of second order linear and nonlinear Ordinary differential equations with known symmetry.
Subject Classification:[2010]76M60, 58J70.

Keywords:
Lie symmetry, Second order*ODE*, SageMath software, Infinitesimal.

1 Introduction

The symmetry approach is an effective tool for solving ordinary and partial differential equations [3, 4, 6, 12, 13, 14, 15, 17, 19).]. In case of ordinary differential equation of second order, $y''=w(x,y,y')$ with known symmetries, associated with it, the canonical coordinates $r=r(x,y), s=s(x,y)$ can be found which reduce the order of second order ordinary differential equation by one and transform the equation in first order ODE.
In this paper we have prepared a program in SageMath [5, 9, 10, 11, 20] for reducing the order of Second order linear and nonlinear ODE with known symmetries . The table is given with 15 different types of ODE with known symmetries given by [8].
The program takes input as $w(x,y,y')$ of ODE $y''=w(x,y,y')$ with number of type (1-15) of ODE mentioned in the table below and transform the equation in variable v and r of first order ODE as explained with examples in the paper.
If the type of second order $y''=w(x,y,y')$ is not included in the table then by giving input number 16, value of $w(x,y,y')$ and the values of $x=x(r,s), y=y(r,s)$ one can find reduced ODE $y''=w(x,y,y')$ as explained with the example.
Since SageMath is a free open-source mathematics software system covered by the GPL, anyone can use the programme provided in the paper, which is very helpful for researchers working with linear/nonlinear ODE.
The program's source code can be downloaded from the provided link https://rb.gy/d1187. **and can be run using SageMath cloud or SageMath cell with link** http://sagecell.sagemath.org/ **without installing SageMath on computer**.

2 Symmetry of second order ODE

Consider $y''=W(x,y,y')$ the Second order ODE with associate infinitesimal operator $X=\xi(x,y)\frac{\partial}{\partial x}+\eta(x,y)\frac{\partial}{\partial y}$. The infinitesimal $X=\xi(x,y)\frac{\partial}{\partial x}+\eta\frac{\partial}{\partial y}$ is associated with the ODE $y''=W(x,y,y')$ if $X^{(2)}(y''-W(x,y,y'))=0$ when $y''=W(x,y,y')$. This is called linearized symmetry condition for second order ODE [2, 7, 16, 18] where

$$\begin{aligned} X^{(2)} &= \xi(x,y)\partial_x+\eta\partial_y+\eta^{(1)}\partial_{y'}+\eta^{(2)}\partial_{y''} \\ \eta^{(1)} &= \eta_x-y'(\xi_x-\eta_y)-y'^2\xi_y \\ \eta^{(2)} &= \eta_{xx}-y'(\xi_{xx}-2\eta_{xy})-y'^2(2\xi_{xy}-\eta_{yy})-y'^3\xi_{yy}-\{2\xi_x+3\xi_y y'-\eta_y\}y''. \end{aligned} \quad (2.1)$$

The canonical coordinates $r=r(x,s), s=r(x,s)$ can be found from equations [1]

$$\xi(x,y)\frac{\partial s}{\partial x}+\eta(x,y)\frac{\partial s}{\partial y}=1$$
$$\xi(x,y)\frac{\partial r}{\partial x}+\eta(x,y)\frac{\partial r}{\partial y}=0. \quad (2.2)$$

Using the relation $r=r(x,s), s=r(x,s)$ we find $x=\phi(r,s)$ and $y=\psi(r,s)$.
Using these relation we obtain

$$y'=\frac{dy}{dx}=\frac{D_r(y)}{D_r(x)}=\frac{\frac{\partial\psi}{\partial r}+s\prime\frac{\partial\psi}{\partial s}}{\frac{\partial\phi}{\partial r}+s\prime\frac{\partial\phi}{\partial r}}=H(r,s,s') \quad (2.3)$$

$$y''=\frac{d^2y}{dx^2}=\frac{D_r(y\prime)}{D_r(x)}=\frac{\frac{\partial H}{\partial r}+s\prime\frac{\partial H}{\partial s}+s\prime\prime\frac{\partial H}{\partial s\prime}}{\frac{\partial\phi}{\partial r}+s\prime\frac{\partial\phi}{\partial s}+s\prime\prime\frac{\partial\phi}{\partial s\prime}} \quad (2.4)$$

where $D_r=\frac{\partial}{\partial r}+s'\frac{\partial}{\partial s}+s''\frac{\partial}{\partial s\prime}+\cdots$ From (2.3) , (2.4) and $y''=w(x,y,y')$ we obtain

$$\frac{\frac{\partial H}{\partial r}+s\prime\frac{\partial H}{\partial s}+s\prime\prime\frac{\partial H}{\partial s\prime}}{\frac{\partial\phi}{\partial r}+s\prime\frac{\partial\phi}{\partial s}+s\prime\prime\frac{\partial\phi}{\partial s\prime}}=w(\phi,\psi,H) \quad (2.5)$$

Solving the equation for s'' we obtain [7]

$$s''=\theta(r,s,s') \quad (2.6)$$

The ODE, however, is invariant under the group of translation in the s direction since r,s are canonical coordinates,therefore $\frac{\partial}{\partial s}\theta(r,s,s')=0$ which implies $\theta(r,s,s')=\theta(r,s')$. Therefore from (2.6) we have $s''=\theta(r,s')$. By taking $v=s'$ we obtain first order ODE $v_r=\theta(r,v)$ where $v_r=\frac{dv}{dr}$.
If the reduced equation has general solution

$$v=G(r,c_1). \quad (2.7)$$

then the ODE has general solution

$$s(x,y)=\int r(x,y)\,G(r,c_1)dr+c_2 \quad (2.8)$$

Using relations $x=\phi(r,s), y=\psi(r,s)$ we obtain general solutions in terms of x and y of the second order ordinary differential equation $y''=w(x,y,y')$.

3 Tables for types of ODE with canonical coordinates

In the book Lie group analysis of differential equations volume 1 Ibragimov [8] has given classification of second order ODE with known symmetries. Using infinitesimal $X=\xi(x,y)\frac{\partial}{\partial x}+\eta\frac{\partial}{\partial y}$ associated with ODE we obtained r,s the canonical coordinates solving equations 2.2. Following table gives the type of equations with infinitesimals associated with ODE and values of x,y in terms of r,s .

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, No. 2, September : 2023

Type of equation	Second order ODE	infinitesimal oprator	Canonical coordinates,r,s,in terms of x,y
1	y"=F(y,y')	X=$\frac{\partial}{\partial x}$	x=s, y=r
2	y"=F(x,y')	X=$\frac{\partial}{\partial y}$	x=r, y=s
3	$y'' = \left(\frac{y'^3}{x^3}\right) F\left(\frac{y}{x}, \frac{y - xy'}{y'}\right)$	X=$\frac{\partial}{\partial y}$	$x = -\frac{1}{rs}, y = -\frac{1}{s}$
4	$y'' = y'^3 F\left(y, \frac{y - xy'}{y'}\right)$	X=$y\frac{\partial}{\partial x}$	$x = rs, y = r$
5	$y'' = F(x, -xy' + y)$	X=$x\frac{\partial}{\partial x}$	$x = r, y = rs$
6	$y'' = \left(\frac{1}{x^3}\right) F\left(\frac{y}{x}, y - xy'\right)$	X=$x^2\frac{\partial}{\partial x} + xy\frac{\partial}{\partial y}$	$x = r, y = rs$
7	$y'' = \left(\frac{1}{x}\right) F\left(\frac{y}{x}, y'\right)$	X=$x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}$	$x = e^s, y = re^s$
8	$y'' = \frac{1}{x}\left(x^2 y'^3 F\left(\frac{y}{x}, y - xy'\right) - y'\right)$	X=$xy\frac{\partial}{\partial x}$	$x = e^{rs}, y = r$
9	$y'' = F\left(ye^{-x}, \frac{y'}{y}\right)$	X=$\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}$	$x = s, y = re^s$
10	$y'' = yF\left(x, \frac{y'}{y}\right)$	X=$y\frac{\partial}{\partial y}$	$x = r, y = e^s$
11	$y'' = \frac{1}{y}\left(y'^2 + y^2 F\left(x, \frac{xy'}{y} - \ln(y)\right)\right)$	X=$xy\frac{\partial}{\partial y}$	$x = r, y = e^{rs}$
12	$y'' = F(kx + ly, y')$	X=$l\frac{\partial}{\partial y} - k\frac{\partial}{\partial y}$	$x = e^{ls}, y = r - ks$
13	$y'' = \frac{1}{p(x)}(p''(x)y + F(x, p(x)y' - p'(x)y))$	X=$p(x)\frac{\partial}{\partial y}$	$x = r, y = sp(r)$
14	$y'' = x^{(k-2)} F\left(\frac{y}{x^k}, x^{(1-k)y'}\right)$	X=$x\frac{\partial}{\partial y} + ky\frac{\partial}{\partial y}$	$x = e^s, y = re^{ks}$
15	$y'' = -\frac{q''(y)}{q(y)} xy'^3 + y'^3 F\left(y, -\frac{1}{y'} - x\frac{q'(y)}{q(y)}\right)$	X=$q(y)\frac{\partial}{\partial x}$	$x = sq(r), y = r$

4 Codes of Program in SagMath

Program in Sage-Math to reduce order of second order ODE with known symmetries

```
print(' Program is to solve diff equ of the type y_xx=w(x,y,y_x)')
var('x,y,r,s,v,s_r,v_r,y_x,y_xx,s_rr,k,l,q_xx')
function('F')
function('q')
function('p')
import math
@interact
def change_varible(A=input_box(default=(1),label='Insert type (1-16) of diff equ')):
  U=[F(y,y_x),F(x,y_x),(y_x^3/x^3)*F(y/x,(y-x*y_x)/y_x),(y_x)^3*F(y,(y-x*y_x)/y_x),
     F(x,y-x*y_x),(1/x^3)*F(y/x,y-x*y_x),(1/x)*F(y/x,y_x),(1/x)*(-y_x+x^2*y_x^3*F(y,(y)/(x*y_x)-
ln(x))),y*F(y*e^(-x),y_x/y),
y*F(x,y_x/y),(1/y)*(y_x^2+y^2*F(x,(x*y_x/y)-ln(y)))]
  if(A==12):
   @interact
   def R_S12 (w=input_box(default=F(2*x+3*y,y_x),label='Insert function
w(x,y,y_x)'),S=input_box(default=(2),label='Insert value of
k'),R=input_box(default=(3),label='Insert value of l')):
     l=R
     k=S
```

```
      H1=((diff(r-k*s,r)+diff(r-k*s,s)*s_r)/(diff(l*s,r)+diff(l*s,s)*s_r))
      G=w(x=l*s,y=r-k*s,y_x=H1)

H2=solve((diff(H1,r)+diff(H1,s)*s_r+diff(H1,s_r)*s_rr)/(diff(l*s,r)+diff(l*s,s)*s_r+diff(l*s,s_r)*s_rr
)==G,s_rr)
      H4=H2[0].rhs()
      H3=H2[0].lhs()
      H4=H4(s_r=v)
      H3=H3(s_rr=v_r)
      print('Diff Equation is of the type y_xx=w(x,y,y_x), where w(x,y,y_x)=',F(k*x+l*y,y_x))
      print('Using canonical coordinates,','x=',l*s,'y=',r-k*s,',equation transformed to')
      M=H3==H4
      Z=M.simplify_full()
      show(Z)
      print('which is first order ODE in v and r, where v resprests s_r and v_r represents derivative of
v w.r.to r')
  elif(A==13):
    @interact
    def R_S13 (w=input_box(default=[(1/p(x))*(diff(p(x),x,x)*y+F(x,p(x)*y_x-
diff(p(x),x)*y)),p(x)],label='Insert function w(x,y,y_x),Insert value of P(x)'),
          ):
      S1=w
      S2=S1[0]
      P=S1[1](x=r)
      H1=((diff(s*P,r)+diff(s*P,s)*s_r)/(diff(r,r)+diff(r,s)*s_r))
      G=S2(x=r,y=s*P,y_x=H1)
      print("""Diff Equation is of the type y_xx=w(x,y,y_x),
          where,""",'w(x,y,y_x)=',S2)

H2=solve((diff(H1,r)+diff(H1,s)*s_r+diff(H1,s_r)*s_rr)/(diff(r,r)+diff(r,s)*s_r+diff(r,s_r)*s_rr)==G,
s_rr)
      H4=H2[0].rhs()
      H3=H2[0].lhs()
      H4=H4(s_r=v)
      H3=H3(s_rr=v_r)
      print('Using canonical coordinates,','x=',r,'y=',s*P,',equation transformed to')
      M=H3==H4
      Z=M.simplify_full()
      show(Z)
      print('which is first order ODE in v and r, where v resprests s_r and v_r represents derivative of
v w.r.to r')
  elif(A==14):
    @interact
    def R_S14 (w=input_box(default=x^(k-2)*F(x^(-k)*y,x^(1-k)*y_x),label='Insert function
w(x,y,y_x)'),
S=input_box(default=(k),label='Insert value of k')):

k=S
```

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, No. 2, September : 2023

```
H1=((diff(r*e^(s*k),r)+diff(r*e^(s*k),s)*s_r)/(diff(e^s,r)+diff(e^s,s)*s_r))G=w(x=e^s,y=r*e^(s*k),y
_x=H1)

H2=solve((diff(H1,r)+diff(H1,s)*s_r+diff(H1,s_r)*s_rr)/(diff(e^s,r)+diff(e^s,s)*s_r+diff(e^s,s_r)*s_r
r)==G,s_rr)
      H4=H2[0].rhs()
      H3=H2[0].lhs()
      H4=H4(s_r=v)
      H3=H3(s_rr=v_r).simplify_full()
      print('Diff Equation is of the type y_xx=w(x,y,y_x), where w(x,y,y_x)=',x^(k-2)*F(x^(-
k)*y,x^(1-k)*y_x))
      print('Using canonical coordinates,','x=',e^s,'y=',r*e^(s*k),',equation transformed to')
      M=H3==H4
      Z=M.simplify_full()
      show(Z)
      print('which is first order ODE in v and r, where v resprests s_r and v_r represents derivative of
v w.r.to r')

  elif(A==15):
    @interact
    def R_S15 (w=input_box(default=[-(diff(diff(q(y),y),y)/q(y))*x*y_x^3+(y_x)^3*F(y,1/(y_x)-
(diff(q(y),y)*x/q(y))),q(y)],label='Insert function w(x,y,y_x),Insert value of q(y)'),
            ):
      S1=w
      S2=S1[0]
      Q=S1[1](y=r)
      H1=((diff(r,r)+diff(r,s)*s_r)/(diff(s*Q,r)+diff(s*Q,s)*s_r))
      G=S2(x=s*Q,y=r,y_x=H1)
      print("""Diff Equation is of the type y_x=w(x,y,y_x),
           where,""",'w(x,y,y_x)=',S2)

H2=solve((diff(H1,r)+diff(H1,s)*s_r+diff(H1,s_r)*s_rr)/(diff(s*Q,r)+diff(s*Q,s)*s_r+diff(s*Q,s_r)*
s_rr)==G,s_rr)
      H4=H2[0].rhs()
      H3=H2[0].lhs()
      H4=H4(s_r=v)
      H3=H3(s_rr=v_r)
      print('Using canonical coordinates,','x=',s*Q,'y=',r,',equation transformed to')
      M=H3==H4
      Z=M.simplify_full()
      show(Z)
      print('which is first order ODE in v and r, where v resprests s_r and v_r represents derivative of
v w.r.to r')

  elif(A>15):
    @interact
```

```
    def R_S16 (w=input_box(default=[(3/x-2*x)*y_x+4*y,r,e^s],label='Insert values of
w(x,y,y_x),x=x(r,s),y=y(r,s)'),
          ):
      print('[Insert values of w(x,y,y_x),and x=x(r,s),y=y(r,s) in terms of canonical coordinates r,s]')
      S1=w
      S2=S1[0]
      S3=S1[1]
      S4=S1[2]
      Q=S1[1](y=r)
      H1=((diff(S4,r)+diff(S4,s)*s_r)/(diff(S3,r)+diff(S3,s)*s_r))
      G=S2(x=S1[1],y=S1[2],y_x=H1)
      print("""Diff Equation is of the type y_x=w(x,y,y_x),
          where,""",'w(x,y,y_x)=',S2)

H2=solve((diff(H1,r)+diff(H1,s)*s_r+diff(H1,s_r)*s_rr)/(diff(S3,r)+diff(S3,s)*s_r+diff(S3,s_r)*s_rr)
==G,s_rr)
      H4=H2[0].rhs().simplify_full()
      H3=H2[0].lhs()
      H4=H4(s_r=v)
      H3=H3(s_rr=v_r)
      print('Using canonical coordinates,','x=',s*Q,'y=',r,',equation transformed to')
      M=H3==H4
      Z=M.simplify_full()
      show(Z)
      print('which is first order ODE in v and r, where v resprests s_r and v_r represents derivative of
v w.r.to r')

  else:
    @interact
    def R_S1 (w=input_box(default=U[A-1],label='Insert value of w(x,y,y_x)')):
      print('Diff Equation is of the type y_xx=w(x,y,y_x), where w(x,y,y_x)=',U[A-1])
      L=[[s,r],[r,s],[-1/(r*s),-1/s],[r*s,r],
        [r,r*s],[-1/s,-r/s],[e^s,r*e^s],[e^(r*s),r],[s,r*e^s],[r,e^s],
        [r,e^(r*s)]]
      H1=((diff(L[A-1][1],r)+diff(L[A-1][1],s)*s_r)/(diff(L[A-1][0],r)+diff(L[A-1][0],s)*s_r))
      G=w(x=L[A-1][0],y=L[A-1][1],y_x=H1)
      H2=solve((diff(H1,r)+diff(H1,s)*s_r+diff(H1,s_r)*s_rr)/
       (diff(L[A-1][0],r)+diff(L[A-1][0],s)*s_r+diff(L[A-1][0],s_r)*s_rr)==G,s_rr)
      H4=H2[0].rhs()
      H3=H2[0].lhs()
      H4=H4(s_r=v)
      H3=H3(s_rr=v_r)
      print('Using canonical coordinates,','x=',L[A-1][0],'y=',L[A-1][1],',equation transformed to')
      M=H3==H4
      Z=M.simplify_full()
      show(Z)
      print('which is first order ODE in v and r, where v resprests s_r and v_r represents derivative of v
w.r.to r')
```

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, No. 2, September : 2023

5 Examples

Example 5.1 *Let*

$$y'' = F(y, y') \tag{5.1}$$

which is equation of type(1) in above table. By giving **inputs** as 1, and value of w(x,y,y_x)=F(y,y_x) we get **output** as ODE of order one in variable v and r .

$$v_r = -v^3 F(r, \frac{1}{v}) \tag{5.2}$$

where $v = \frac{ds}{dr}$ and v_r is derivative of v w.r.to r.

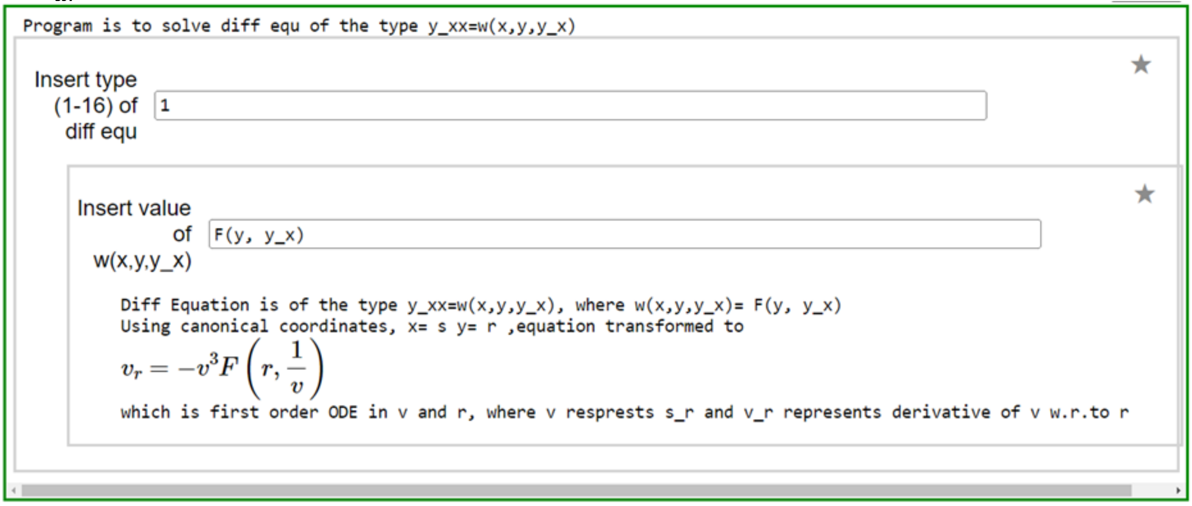

Figure 1: Reduced first order ODE

Example 5.2 *Let*

$$y'' = F\left(y, \frac{y - xy'}{y'}\right) y'^3 \tag{5.3}$$

which is equation of type(4) in above table. By giving **inputs** as 4, and value of w(x,y,y_x)= F(y, -(x*y_x - y)/y_x)*y_x^3, we get **output** as ODE of order one in variable v and r .

$$v_r = -\frac{2v + F(r, r^2 v)}{r} \tag{5.4}$$

where $v = \frac{ds}{dr}$ and v_r is derivative of v w.r.to r.

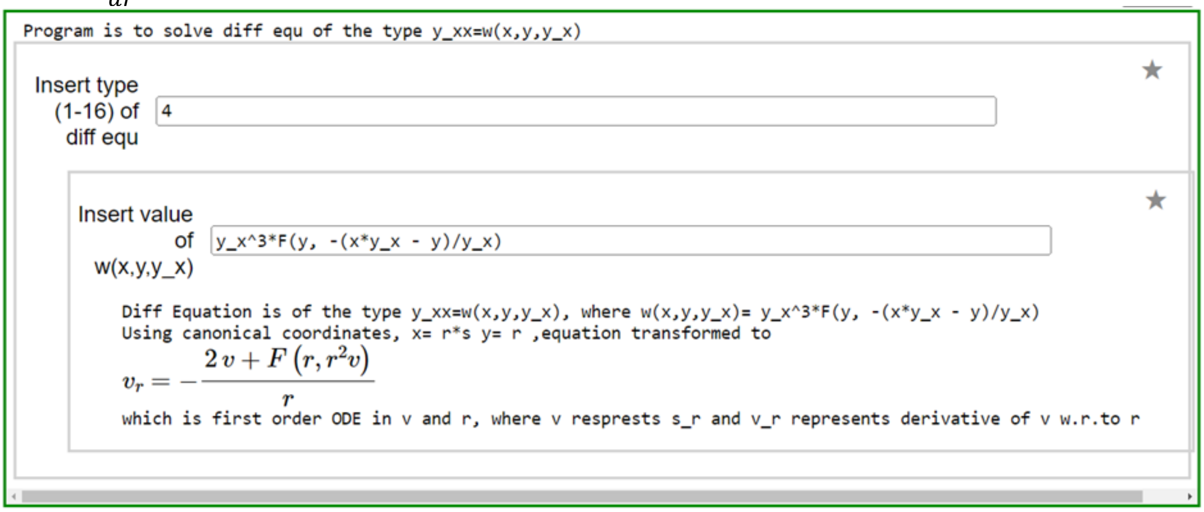

Figure 2: Reduced first order ODE

Now let $F\left(y, \frac{y-xy'}{y'}\right) = y + \frac{y-xy'}{y'}$. By giving input as

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, No. 2, September : 2023

w(x,y,y_x)= y_x^3*(y+(y-x*y_x)/y_x) we get output as reduced ODE of order one in terms of variable v and r.

$$v_r = -\frac{r+(r^2+2)v}{r} \tag{5.5}$$

where $v = \frac{ds}{dr}$ and v_r is derivative of v w.r.to r.

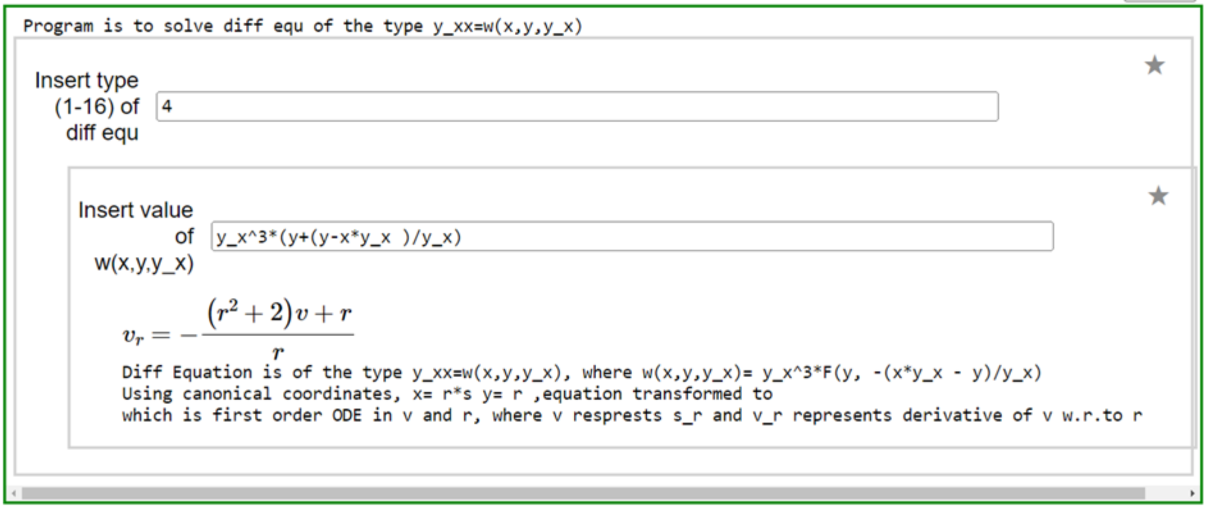

Figure 3: Reduced first order ODE

The equation (5.5) can be further solved to obtain

$$v = \frac{ds}{dr} = \left(\frac{2}{r^2} - 1\right) + c\frac{1}{r^2}e^{-r^2/2} \tag{5.6}$$

which in variable separable form on solving and using relation $x = rs, y = r$ we get in terms of x,y the solution ODE .

Example 5.3 Let[1] $y'' = -\frac{2xy'^3}{y^2}$ (5.7)

The type of this ODE is not included in above table This ODE includes eight parameter family of infinitesimals.

Taking infinitesimal operator $X = x\frac{\partial}{\partial x}$ admitted by ODE, we obtain $x = e^s, y = r$.

By giving **inputs** as 16, values of w(x,y,y_x)=-2*x*y_x^3/y^2 and $x = e^s, y = r$ and pressing **(shift+enter)** keys we get **output** as first order ODE in v and r .

$$v_r = -\frac{r^2v^2-2}{r^2} \tag{5.8}$$

where $v = \frac{ds}{dr}$ and v_r is derivative of v w.r.to r.

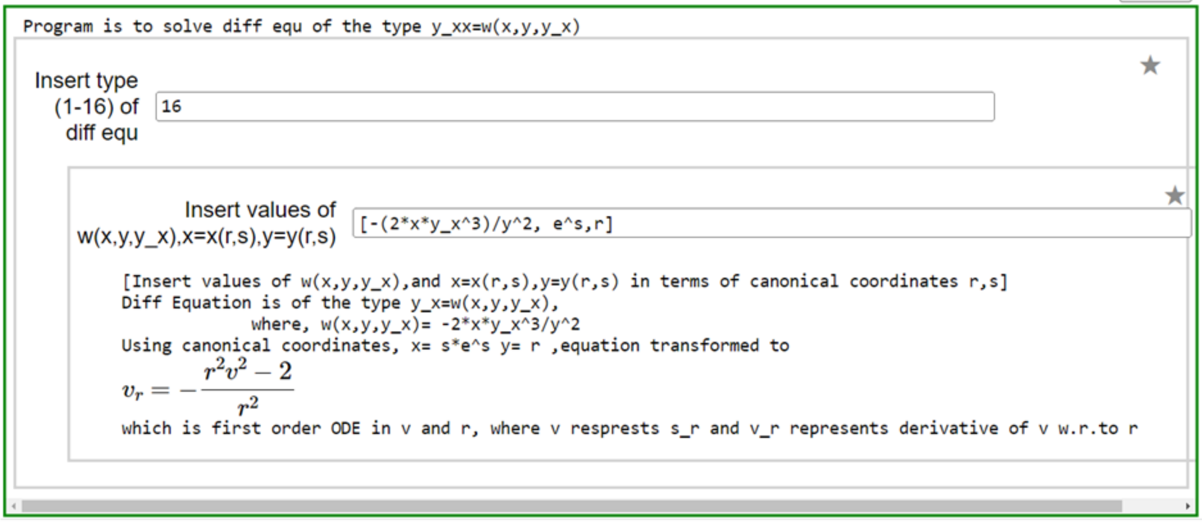

Figure 4: Reduced first order ODE

Now take infinitesimal operator $X = y\frac{\partial}{\partial y}$, we obtain $x = e^s, y = r$.

By giving **inputs** as 16, values of w(x,y,y_x)=-2x*y_x^3/y^2 and $x = r, y = e^s$ and pressing **(shift+enter)** keys we get **output** as first order ODE in v and r .

$$v_r = -2rv^3 - v^2 \tag{5.9}$$

where $v = \frac{ds}{dr}$ and v_r is derivative of v w.r.to r.

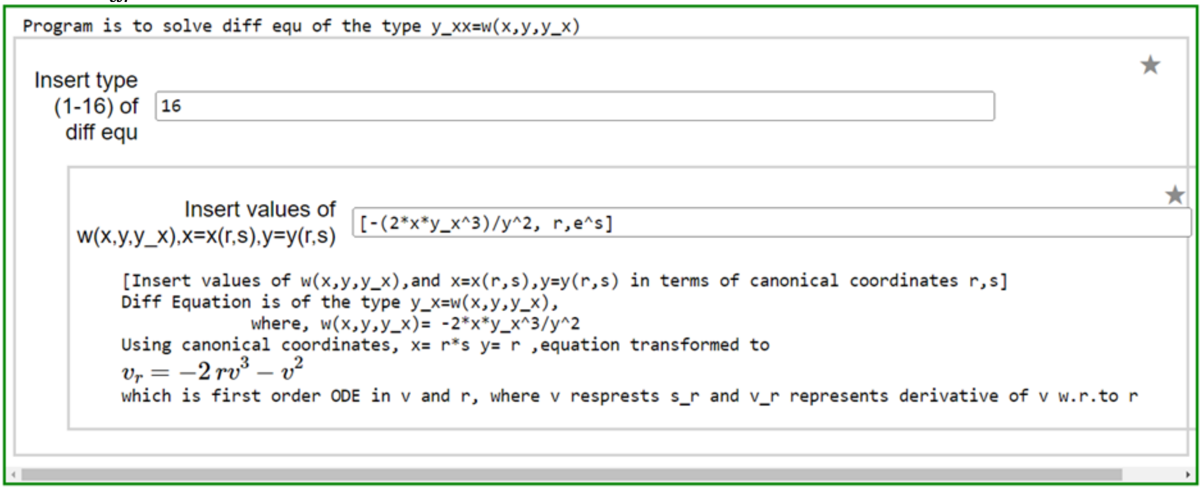

Figure 5: Reduced first order ODE

Remark 5.1 *Similarly as explained in above examples by giving the input as type of equation and value of $w(x, y, y')$ by considering the particular value of F involve in $w(x, y, y')$ of ODE $y'' = w(x, y, y')$ we obtain reduced ODE of order one.*

6 Conclusion

The program given in the paper is very useful to reduce the order of second order ODE by one with known symmetry. The codes of the program can be further extended to include higher order ODEs.

References

[1] Arigo, D. J. (2015). Symmetry Analysis of Differential Equations An Introduction. John Wiley and Sons.

[2] Bluman, G. and Kumei, S. (1989). Symmetry and Differential equations. Springer-Verlag New

York, Inc.
[3] Can, M. (1996). Lie Symmetries Of Differential Equations By Computer Algebra, Mathematical and Computational Applications. 1(1).
[4] Dresner, L. (1998). Applications of Lie's Theory of Ordinary and Partial Differential Equations. CRC Press.
[5] Finch, C. (2011). Sage Beginner's Guide. Packt Publishing Ltd.
[6] Hansen, A. (1964). Similarity Analyses of Boundary Value Problems in Engineering. Prentice-Hall Inc.
[7] Hydon, P. E. (2005). Symmetry Methods for Differential Equations. Cambrige University Press.
[8] Ibragimov, N. (1994). CRC Handbook of Lie Group Analysis of Dif- ferential Equations Symmetries, Exact Solutions, and Conservation Laws, Volume I. CRC Press.
[9] Joyner, D. (2012). Introduction to Differential Equations Using Sage. Johns Hopkins University Press.
[10] Kosan, T. (2008). Sage for Newbies.
[11] Mezei, R. A. (2016). An Introduction to SAGE Programming with Ap- plication to SAGE Interacts for Numerical Methods. John Wiley and Sons.
[12] Moran, M. and Gaggioli (1968). Similarity analyses via group theory.A.I.A.A. Journal, 6(10).
[13] Na, T. Y. and Hansen, A. G. (1971). Similarity Analysis of Differential Equations by Lie Group. Journal of Franklin Institute, 292(6).
[14] Olver, P. (1993). Applications of Lie groups to differential equations (2nd ed.), Graduate Texts in Mathematics. Springer-Verlag, New York.
[15] Pulov, V. I., Chacarov, E. J., and Uzunov, I. M. (2007). A Computer Algebra Application To Determination Of Lie Symmetries Of Partial Dif- ferential Equations. Serdica Journal of Computing, 1:505–518.
[16] Ross, S. L. (2018). Differential equation. Wiley.
[17] Seshadri, R. and Na, T. (1985). Group invariance in engineering bound-*ary value problems*. Springer-Verlag.
[18] Steinhour, R. A. (2013). *The Truth About Lie Symmetries: Solving Differential Equation with Symmetry Methods*. The College of Wooster Libraries Open Works.
[19] Stephani, H. (1989). Differential equations their solutions using sym- metry. Cambridge University Press.
[20] Zimmermann, P. (2018). Computational Mathematics with SageMath .