Print of all equations for FIDISOL/CADSOL

Clear Variables

Input Equations of Motion

Modification to the functions

Here we will write the *ansatz* function as FIDISOL/CADSOL wants. We implement as well the radial coordinate transformation $\bar{x} = \frac{x}{1+x}$ which changes the first and second radial derivative of the functions.

Ansatz functions

In[ ]:=
```
F1[x, θ] = U1;
F2[x, θ] = U2;
F0[x, θ] = U3;
W[x, θ] = U4;
```

In[ ]:=
```
F1^(1,0)[x, θ] = ff^2 U1x;
F2^(1,0)[x, θ] = ff^2 U2x;
F0^(1,0)[x, θ] = ff^2 U3x;
W^(1,0)[x, θ] = ff^2 U4x;
```

In[ ]:=
```
F1^(2,0)[x, θ] = ff^4 U1xx - 2 ff^3 U1x;
F2^(2,0)[x, θ] = ff^4 U2xx - 2 ff^3 U2x;
F0^(2,0)[x, θ] = ff^4 U3xx - 2 ff^3 U3x;
W^(2,0)[x, θ] = ff^4 U4xx - 2 ff^3 U4x;
```

In[ ]:=
```
F1^(1,1)[x, θ] = ff^2 U1xy;
F2^(1,1)[x, θ] = ff^2 U2xy;
F0^(1,1)[x, θ] = ff^2 U3xy;
W^(1,1)[x, θ] = ff^2 U4xy;
```

In[ ]:=
```
F1^(0,1)[x, θ] = U1y;
F2^(0,1)[x, θ] = U2y;
F0^(0,1)[x, θ] = U3y;
W^(0,1)[x, θ] = U4y;
```

In[ ]:=
```
F1^(0,2)[x, θ] = U1yy;
F2^(0,2)[x, θ] = U2yy;
F0^(0,2)[x, θ] = U3yy;
W^(0,2)[x, θ] = U4yy;
```

In[ ]:=
```
H[x] := H;
H'[x] := derH;
H''[x] := der2H;
S[x] := S;
S'[x] := 2 x;
S''[x] := 2;
```

Trigonometric functions

In[ ]:=
```
Unprotect[Sin];
Unprotect[Cos];
Unprotect[Cot];
Unprotect[Csc];
Sin[θ] = sn;
Cos[θ] = cs;
Cos[2 θ] = cs2;
Cot[θ] = cs / sn;
Csc[θ] = csc;
```

Convert E**x to dExp(x)

In[ ]:=
```
Unprotect[Power];
Power /: Format[Power[E, x_], FortranForm] := dExp[x]
Protect[Power];
```

Set of the width of the output

For some reason, Fortran does not like very long expressions.

In[ ]:=
```
SetOptions[$Output, PageWidth → 65];
```

Print Equations

Equation 1:

```
In[ ]:= FortranForm[Simplify[Eq1]]
Out[ ]//FortranForm=
      (sn*(-((dExp(2*U2 - 2*U3)*S**2*sn**3*
     -           (ff**4*H*S*U4x**2 + U4y**2))/x) -
     -      2*H*(4*ff**3*H*S*sn*U1x*x -
     -         2*ff**4*H*S*sn*(U1xx - U2x*U3x)*x +
     -         2*(-(sn*U1yy) + cs*U3y + sn*U2y*U3y)*x +
     -         ff**2*sn*(2*H*S*U2x + derH*S*(-U1x + U2x)*x -
     -            2*H*(U1x + U2x - U3x)*x**2))))/(4.*H**2*S)
```

Equation 2:

```
In[ ]:= FortranForm[Simplify[Eq2]]
Out[ ]//FortranForm=
      (sn*(dExp(2*U2 - 2*U3)*S**3*sn**3*
     -       (ff**4*H*S*U4x**2 + U4y**2) +
     -      2*H*x*(-2*ff**3*H*S**2*sn*U2x*x +
     -         ff**4*H*S**2*sn*(U2x**2 + U2xx + U2x*U3x)*x +
     -         ff**2*S*sn*(H*S*U2x + derH*S*U2x*x +
     -            H*(2*U2x + U3x)*x**2) +
     -         x*(cs*S*(2*U2y + U3y) - H*sn*x**2 +
     -            S*sn*(-1 + 2*H + U2y**2 + U2yy + U2y*U3y +
     -               derH*x)))))/(2.*H**2*S**2*x**2)
```

Equation 3:

```
In[ ]:= FortranForm[Simplify[Eq3]]
Out[ ]//FortranForm=
     -((dExp(2*U2 - 2*U3)*S*sn**3*
     -       (ff**4*H*S*U4x**2 + U4y**2))/(H**2*x)) -
     - 4*ff**3*sn*U3x*x +
     - 2*ff**4*sn*(U2x*U3x + U3x**2 + U3xx)*x +
     - (2*ff**2*sn*(S*(U2x + 2*U3x) - U2x*x**2))/S +
     - (derH*sn*(-x**2 + S*(3 + ff**2*(U2x + 3*U3x)*x)))/
     -  (H*S) + (x*(S*(der2H*S*sn + 2*cs*U3y +
     -          2*sn*(U2y*U3y + U3y**2 + U3yy)) +
     -       2*H*sn*(-S + x**2)))/(H*S**2)
```

Equation 4:

In[ ]:=
```
FortranForm[Simplify[Eq4]]
```

Out[ ]//FortranForm=
```
      (sn*x*(-2*ff**3*H*S*sn*U4x*x +
     -     ff**4*H*S*sn*(3*U2x*U4x - U3x*U4x + U4xx)*x +
     -     3*cs*U4y*x + sn*(3*U2y*U4y - U3y*U4y + U4yy)*x -
     -     ff**2*H*sn*U4x*(S - 5*x**2)))/(H*S)
```

Constraint Equation 1:

In[ ]:=
```
FortranForm[Simplify[ConstrainEq1]]
```

Out[ ]//FortranForm=
```
      (dExp(2*U2)*S**3*sn**3*(ff**4*H*S*U4x**2 - U4y**2) -
     -   dExp(2*U3)*H*x*(-4*ff**3*H*S**2*sn*(U2x + U3x)*x +
     -      2*ff**4*H*S**2*sn*
     -       (U2x**2 + U2xx + U3x**2 - 2*U1x*(U2x + U3x) +
     -         U3xx)*x + x*
     -       (der2H*S**2*sn -
     -         2*S*(2*cs*(-U1y + U2y) +
     -            sn*(-1 + 3*H + U2y**2 + U2yy + U3y**2 -
     -               2*U1y*(U2y + U3y) + U3yy)) + 4*H*sn*x**2)\
     -       + 2*ff**2*H*S*sn*
     -       (-2*S*(U1x - U3x) + (U2x - 3*U3x)*x**2) +
     -      derH*S*sn*(-3*x**2 +
     -         S*(3 + ff**2*(-2*U1x + U2x + 3*U3x)*x))))/
     -   (2.*dExp(2*(U1 + U3))*H*S**2*sn*x**2)
```

Constraint Equation 2:

In[ ]:=
```
FortranForm[Simplify[ConstrainEq2]]
```

Out[ ]//FortranForm=
```
      (dExp(2*U2)*ff**2*S**3*sn**3*U4x*U4y +
     -   dExp(2*U3)*x*(derH*S*sn*(U1y - U3y)*x +
     -      2*H*(cs*ff**2*S*(U1x - U2x)*x + 2*sn*U3y*x**2 +
     -         S*sn*(U1y + ff**2*U1y*(U2x + U3x)*x -
     -            ff**2*(U2xy - U1x*U2y + U2x*U2y + U3xy)*x +
     -            U3y*(-1 + ff**2*(U1x - U3x)*x)))))/
     -   (2.*dExp(2*(U1 + U3))*H*S**2*sn*x**2)
```

Ricci Scalar:

```
In[ ]:= FortranForm[Simplify[R]]
```

```
Out[ ]//FortranForm=
      (dExp(2*U2)*S**2*sn**2*U4y**2 -
     -   4*dExp(2*U3)*H**2*x*
     -    (x - 2*ff**3*S*(U1x + U2x + U3x)*x +
     -       ff**4*S*(U1xx + U2x**2 + U2xx + U2x*U3x + U3x**2 +
     -          U3xx)*x + ff**2*
     -        (S*(U2x + 2*U3x) + (U1x + 2*U2x)*x**2)) -
     -   H*(-(dExp(2*U2)*ff**4*S**3*sn**2*U4x**2) +
     -      dExp(2*U3)*x**2*
     -       (4*(-1 + U1yy + U2y**2 + U2yy + U2y*U3y + U3y**2 +
     -            (cs*(2*U2y + U3y))/sn + U3yy) + 2*derH*x) +
     -      2*dExp(2*U3)*S*x*
     -       (der2H*x + derH*
     -          (3 + ff**2*(U1x + 2*U2x + 3*U3x)*x))))/
     -  (2.*dExp(2*(U1 + U3))*H*S*x**2)
```

Kretschmann Scalar:

Only print this one when you need it, because it is a train of an expression.

```
In[ ]:= FortranForm[KretschmanScalar];
```

Print Jacobian

Here we compute the derivatives of the equations of motion w.r.t the several *ansatz* functions and its first and second derivatives.

Equation 1

```
In[ ]:= Eq1Uxx1 = D[Eq1, U1xx];
        Eq1Uxy1 = D[Eq1, U1xy];
        Eq1Uyy1 = D[Eq1, U1yy];
        Eq1Ux1 = D[Eq1, U1x];
        Eq1Uy1 = D[Eq1, U1y];
        Eq1U1 =  D[Eq1, U1];
```

```
In[ ]:= FortranForm[Simplify[Eq1Uxx1]]
```

```
Out[ ]//FortranForm=
      ff**4*sn**2*x
```

```
In[ ]:= FortranForm[Simplify[Eq1Uxy1]]
```

```
Out[ ]//FortranForm=
      0
```

```
In[ ]:= FortranForm[Simplify[Eq1Uyy1]]
Out[ ]//FortranForm=
(sn**2*x)/(H*S)

In[ ]:= FortranForm[Simplify[Eq1Ux1]]
Out[ ]//FortranForm=
(ff**2*sn**2*x*(derH*S + 2*H*(-2*ff*S + x)))/(2.*H*S)

In[ ]:= FortranForm[Simplify[Eq1Uy1]]
Out[ ]//FortranForm=
0

In[ ]:= FortranForm[Simplify[Eq1U1]]
Out[ ]//FortranForm=
0

In[ ]:= Eq1Uxx2 = D[Eq1, U2xx];
        Eq1Uxy2 = D[Eq1, U2xy];
        Eq1Uyy2 = D[Eq1, U2yy];
        Eq1Ux2 = D[Eq1, U2x];
        Eq1Uy2 = D[Eq1, U2y];
        Eq1U2 =  D[Eq1, U2];

In[ ]:= FortranForm[Simplify[Eq1Uxx2]]
Out[ ]//FortranForm=
0

In[ ]:= FortranForm[Simplify[Eq1Uxy2]]
Out[ ]//FortranForm=
0

In[ ]:= FortranForm[Simplify[Eq1Uyy2]]
Out[ ]//FortranForm=
0

In[ ]:= FortranForm[Simplify[Eq1Ux2]]
Out[ ]//FortranForm=
      -(ff**2*sn**2*(derH*S*x +
     -       2*H*(S + ff**2*S*U3x*x - x**2)))/(2.*H*S)

In[ ]:= FortranForm[Simplify[Eq1Uy2]]
Out[ ]//FortranForm=
-((sn**2*U3y*x)/(H*S))

In[ ]:= FortranForm[Simplify[Eq1U2]]
Out[ ]//FortranForm=
      -(dExp(2*U2 - 2*U3)*S*sn**4*(ff**4*H*S*U4x**2 + U4y**2))/
     -  (2.*H**2*x)
```

```
In[ ]:= Eq1Uxx3 = D[Eq1, U3xx];
        Eq1Uxy3 = D[Eq1, U3xy];
        Eq1Uyy3 = D[Eq1, U3yy];
        Eq1Ux3 = D[Eq1, U3x];
        Eq1Uy3 = D[Eq1, U3y];
        Eq1U3 =  D[Eq1, U3];

In[ ]:= FortranForm[Simplify[Eq1Uxx3]]
Out[ ]//FortranForm=
        0

In[ ]:= FortranForm[Simplify[Eq1Uxy3]]
Out[ ]//FortranForm=
        0

In[ ]:= FortranForm[Simplify[Eq1Uyy3]]
Out[ ]//FortranForm=
        0

In[ ]:= FortranForm[Simplify[Eq1Ux3]]
Out[ ]//FortranForm=
        -((ff**2*sn**2*x*(ff**2*S*U2x + x))/S)

In[ ]:= FortranForm[Simplify[Eq1Uy3]]
Out[ ]//FortranForm=
        -((sn*(cs + sn*U2y)*x)/(H*S))

In[ ]:= FortranForm[Simplify[Eq1U3]]
Out[ ]//FortranForm=
                  (dExp(2*U2 - 2*U3)*S*sn**4*(ff**4*H*S*U4x**2 + U4y**2))/
              -   (2.*H**2*x)

In[ ]:= Eq1Uxx4 = D[Eq1, U4xx];
        Eq1Uxy4 = D[Eq1, U4xy];
        Eq1Uyy4 = D[Eq1, U4yy];
        Eq1Ux4 = D[Eq1, U4x];
        Eq1Uy4 = D[Eq1, U4y];
        Eq1U4 =  D[Eq1, U4];

In[ ]:= FortranForm[Simplify[Eq1Uxx4]]
Out[ ]//FortranForm=
        0

In[ ]:= FortranForm[Simplify[Eq1Uxy4]]
Out[ ]//FortranForm=
        0
```

In[ ]:= **FortranForm[Simplify[Eq1Uyy4]]**

Out[ ]//FortranForm=

```
0
```

In[ ]:= **FortranForm[Simplify[Eq1Ux4]]**

Out[ ]//FortranForm=

```
-(dExp(2*U2 - 2*U3)*ff**4*S**2*sn**4*U4x)/(2.*H*x)
```

In[ ]:= **FortranForm[Simplify[Eq1Uy4]]**

Out[ ]//FortranForm=

```
-(dExp(2*U2 - 2*U3)*S*sn**4*U4y)/(2.*H**2*x)
```

In[ ]:= **FortranForm[Simplify[Eq1U4]]**

Out[ ]//FortranForm=

```
0
```

Equation 2

Equation 3

Equation 4