

A tutorial on GiNaCDE-GUI to solve Differential Equations

GiNaCDE-GUI (V1.0.0)

Mithun Bairagi

*Department of Physics, The University of Burdwan, Golapbag 713104,
West Bengal, India*

January 31, 2022

The usage of GiNaCDE-GUI for solving differential equations is described in this tutorial. The following two examples demonstrate the usage of GiNaCDE-GUI to solve differential equations.

1 Example 1

Let us consider the Eckhaus equation

$$iu_t + u_{xx} + 2\left(|u|^2\right)_x u + |u|^4 u = 0. \quad (1)$$

For solving Eq. (1) using GiNaCDE library, the C++ codes are

```
// eckhaus_FIM.cpp
#include <GiNaCDE/GiNaCDE.h>
int main()
{
    const ex u=reader("u"), t=reader("t"), x=reader("x"),
            a=reader("a"), b=reader("b"),k=reader("k");
    depend(u, {t, x});
    const ex pde = I*Diff(u,t,1) + Diff(u,x,2) + 2*u*Diff(u*conjugate(u),x,1)
            + u*u*conjugate(u)*conjugate(u)*u;
    output = mathematica;
    twcPhase=lst{lst{-2*k*a,k},lst{b,a}};
    paraInDiffSolve=lst{};
    filename = "eckhaus_FIM.txt";
    desolve(pde,{u},FIM);
    return 0;
}
```

In the following, we display the screenshots of each step when we implement the above C++ code in GiNaCDE-GUI.

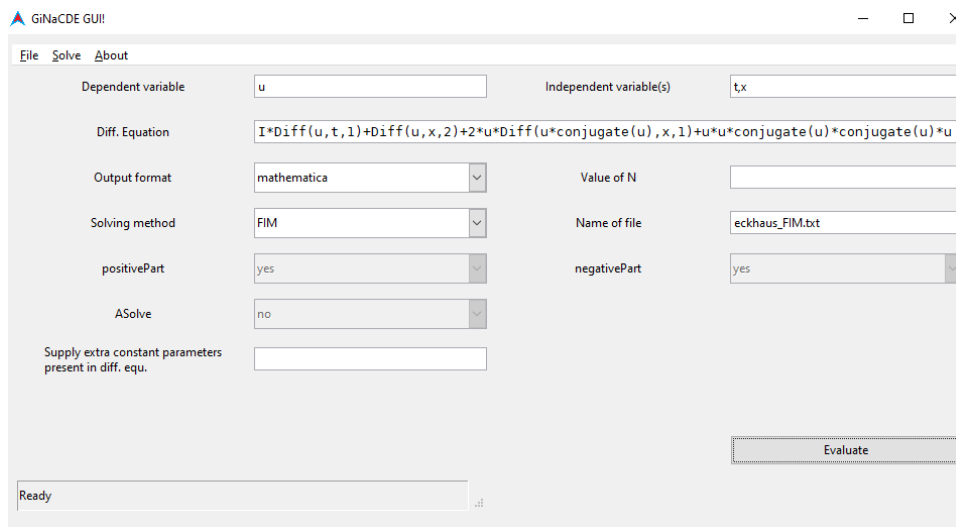


Figure 1: Step 1.

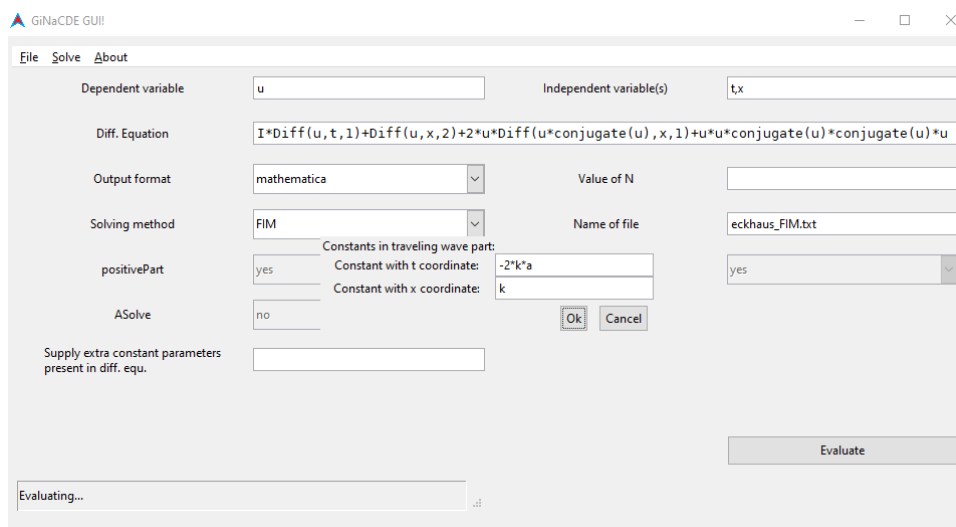


Figure 2: Step 2.

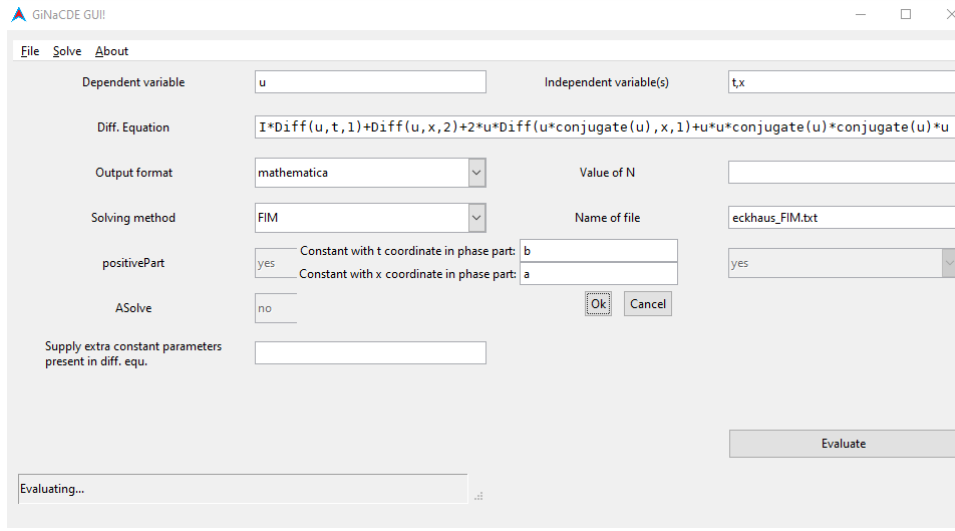


Figure 3: Step 3.

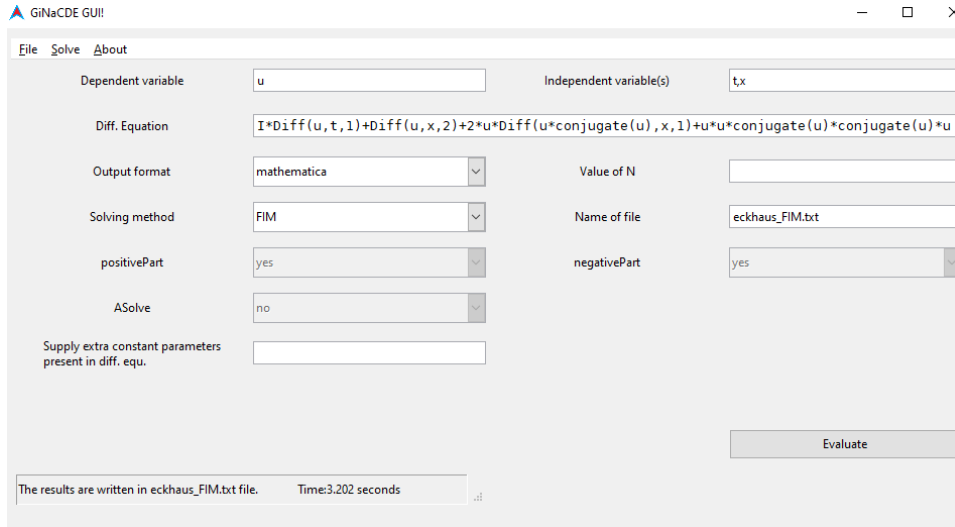


Figure 4: Step 4.

After execution of the last step in fig. 4 the output results are saved in *eckhaus_FIM.txt* file.

2 Example 2

We now discuss an another example considering generalized Camassa-Holm equation

$$u_t + 2ku_x - u_{xxt} + auu_x - 2u_xu_{xx} - uu_{xxx} = 0. \quad (2)$$

The following C++ code solve Eq. (2) applying modified F-expansion method.

```
// Generalized_Camassa-Holm_mF.cpp
#include <GiNaCDE/GiNaCDE.h>
int main()
{
    const ex u=reader("u"),t=reader("t"), x=reader("x"),
        k=reader("k"),a=reader("a"),k_0=reader("k_0"),
        k_1=reader("k_1"),A_1=reader("A_1"),A_3=reader("A_3");
    depend(u, {t,x});
    const ex pde = Diff(u,t,1)+2*k*Diff(u,x,1)-Diff(Diff(u,x,2),t,1)
        +a*u*Diff(u,x,1)-2*Diff(u,x,1)*Diff(u,x,2)
        -u*Diff(u,x,3);

    output = maple;
    twcPhase = lst{lst{k_0,k_1},lst{}};
    degAcoeff = lst{3,0,A_1,0,A_3};
    NValue = 2;
    filename = "Generalized_Camassa-Holm_mF.txt";
    ASolve=false;
    positivePart = true;
    negativePart = true;
    paraInDiffSolve = lst{k,a};
    desolve(pde, {u}, mF_expansion);
    return 0;
}
```

The following screenshots express each step to implement the above C++ code in GiNaCDE-GUI.

The screenshot shows the GiNaCDE GUI with the following settings:

- Dependent variable: `u`
- Independent variable(s): `t,x`
- Diff. Equation: `Diff(u,t,1)+2*k*Diff(u,x,1)-Diff(Diff(u,x,2),t,1)+a*u*Diff(u,x,1)-2*Diff(u,x,1)*Diff(u,x,2)-u`
- Output format: `maple`
- Value of N: `2`
- Name of file: `Generalized_Camassa-Holm_`
- Solving method: `mF_expansion`
- positivePart: `yes`
- negativePart: `yes`
- ASolve: `no`
- Supply extra constant parameters present in diff. equ.: `k,a`

The status bar at the bottom indicates "Ready".

Figure 5: Step 1.

The screenshot shows the same GiNaCDE GUI as Figure 5, but with a dialog box open. The dialog box contains the text "Highest positive integer (delta) of 1st order NLODE (A.E.)" and a text input field containing the value `3`. There are "Ok" and "Cancel" buttons. The status bar at the bottom now indicates "Evaluating...".

Figure 6: Step 2.

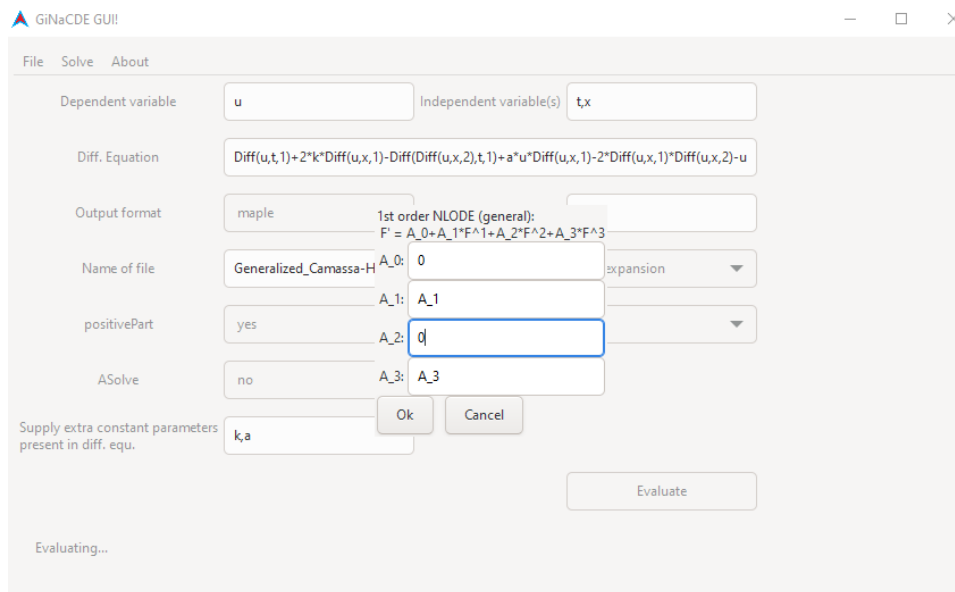


Figure 7: Step 3.

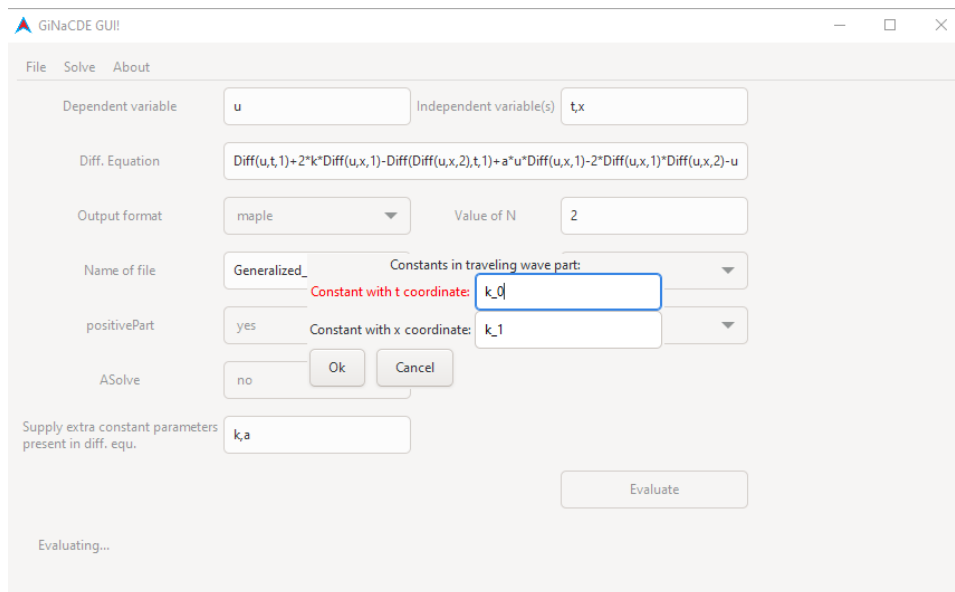


Figure 8: Step 4.

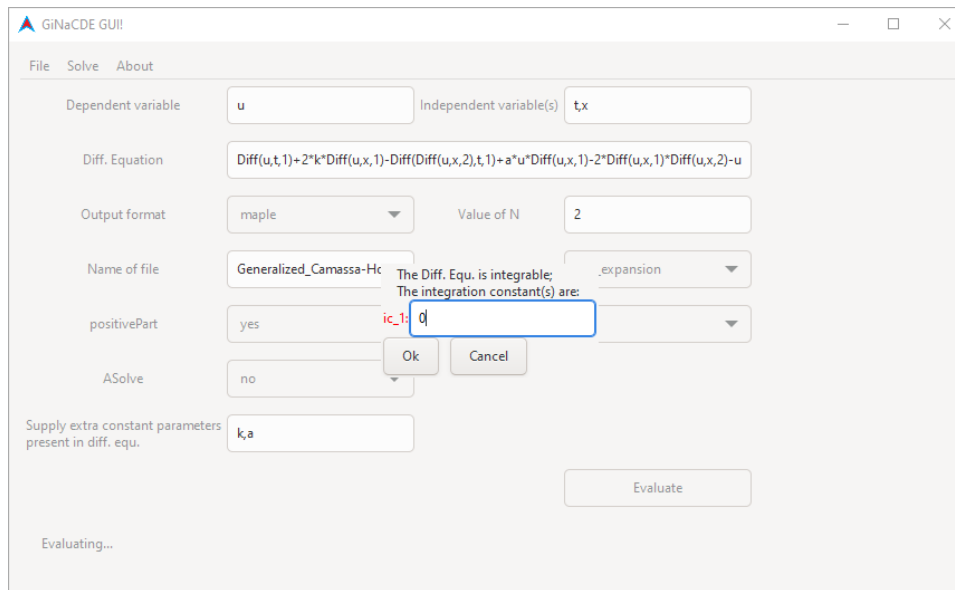


Figure 9: Step 5.

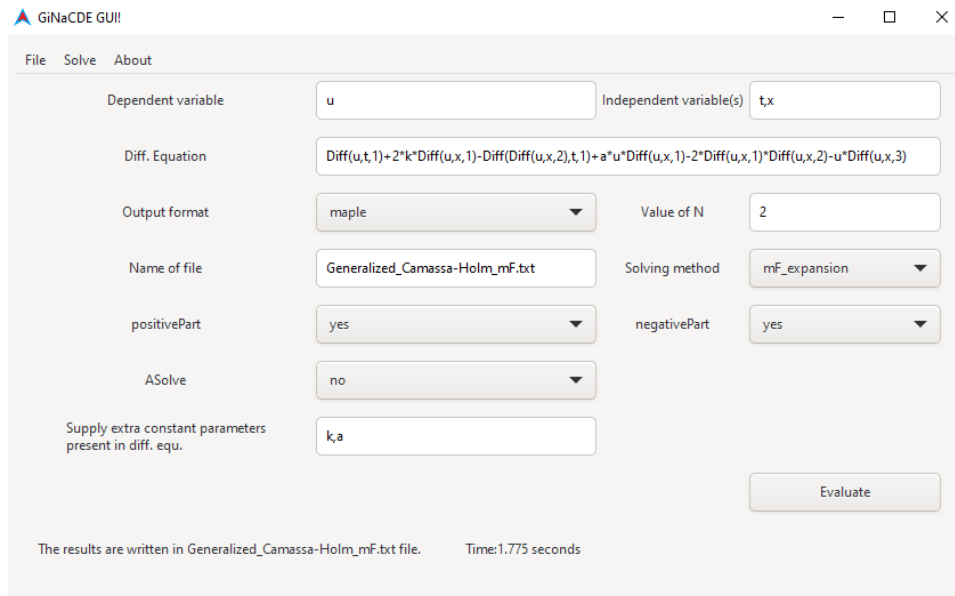


Figure 10: Step 6.

After execution of the last step in fig. 10 the output results are saved in *Generalized_Camassa-Holm.txt* file.