Data for the initial guess - Kerr

Clear Variables

This command clears every variable that is in memory. It will clear variables in all open notebooks.

In[ ]:=
```
ClearAll["Global`*"]
```

Choice of the input parameters

In[ ]:=
```
rH = 0.1;
ΩH = 0.96;
```

Relation to the analytic Kerr Metric

From Appendix A in [*Class. Quant. Grav*., vol. 32, no. 14, p. 144001, 2015.] and Appendix A in the thesis, http://hdl.handle.net/10773/17984, we can compute the physical quantities of a Kerr black hole with the above input parameters

In[ ]:=
```
x = rH ΩH;
```

In[ ]:=

$$\text{ulow[x\_]} := \frac{1}{3}\left(2+\frac{\sqrt{3+x^2}}{x}\,\text{Cos}\left[\frac{1}{3}\left(4\,\pi+\text{ArcCos}\left[\frac{x\,(18+x^2)}{(3+x^2)^{3/2}}\right]\right)\right]\right)$$

In[ ]:=

$$\text{uhigh[x\_]} := \frac{1}{3}\left(2+\frac{\sqrt{3+x^2}}{x}\,\text{Cos}\left[\frac{1}{3}\left(2\,\pi+\text{ArcCos}\left[\frac{x\,(18+x^2)}{(3+x^2)^{3/2}}\right]\right)\right]\right)$$

Let’s consider the solution on the lower branch.

In[ ]:=
```
u = ulow[x]
```

Out[ ]=
```
-0.00966845
```

In[ ]:=
```
ct = rH u;
```

In[∘]:= $M = \frac{1}{2}\left(rH - 2\,ct\right);$

$J = \frac{1}{2}\sqrt{ct\,(ct - rH)}\,\left(rH - 2\,ct\right);$

$AH = 4\,\pi\,(rH - ct)\left(rH - 2\,ct\right);$

$TH = \frac{rH}{4\,\pi\,(rH - ct)\left(rH - 2\,ct\right)};$

In[∘]:=
```
Print["Mass = ", M, "\n", "Angular Momentum = ", J,
 "\n", "Horizon Area = ", AH, "\n", "Hawking Temperature = ", TH]
```

```
Mass = 0.0509668
Angular Momentum = 0.000503565
Horizon Area = 0.129332
Hawking Temperature = 0.773203
```

Grid for the initial solution

xx is the the radial coordinate $\bar{x}$ on the notes. x is the radial coordinate x on the notes.

In[∘]:=
```
nx = 251;
ny = 33;
```

In[∘]:= $xx = \text{Table}\left[i, \left\{i, 0, 1, \frac{1}{(nx - 1)}\right\}\right]$ // N;

$y = \text{Table}\left[i, \left\{i, 0, \frac{\pi}{2}, \frac{\pi/2}{(ny - 1)}\right\}\right]$ // N;

In[∘]:=
```
x = Quiet[xx / (1 - xx)] // N;
x[[nx]] = 10^12;
```

Since when xx=1 --> x -> ∞, we change the last value of the array x, to be a very large number

Computation of the *ansatz* functions

All expression on the Appendices are written in terms of the original radial coordinate r. So we need to convert x back to r.

In[∘]:= $r = \sqrt{x^2 + rH^2};$

```
In[ ]:= F1 = Table[1/2 Log[(1 - ct/r[[j]])^2 + ct (ct - rH) Cos[y[[i]]]^2/r[[j]]^2], {i, 1, ny}, {j, 1, nx}];
        F2 =
          Table[1/2 Log[Exp[-2 F1[[i, j]]] (((1 - ct/r[[j]])^2 + ct (ct - rH)/r[[j]]^2)^2 + ct (rH - ct) (1 - rH/r[[j]]) Sin[y[[i]]]^2/r[[j]]^2)],
            {i, 1, ny}, {j, 1, nx}];
        F0 = -F2;
        W =
          Table[Exp[-2 (F1[[i, j]] + F2[[i, j]])] Sqrt[ct (ct - rH)] (rH - 2 ct) (1 - ct/r[[j]])/r[[j]]^3, {i, 1, ny}, {j, 1, nx}];

In[ ]:= Functions = Table[0, {i, 1, 4 nx ny + 1}];

In[ ]:= F1Reshape = ArrayReshape[F1, {nx ny}];
        F2Reshape = ArrayReshape[F2, {nx ny}];
        F0Reshape = ArrayReshape[F0, {nx ny}];
        WReshape = ArrayReshape[W, {nx ny}];

In[ ]:= For[i = 1, i ≤ 4 nx ny, i++,
         If[Mod[i, 4] == 1,
          Functions[[i]] = F1Reshape[[IntegerPart[(i - 1) / 4] + 1]]];
         If[Mod[i, 4] == 2,
          Functions[[i]] = F2Reshape[[IntegerPart[(i - 1) / 4] + 1]]];
         If[Mod[i, 4] == 3,
          Functions[[i]] = F0Reshape[[IntegerPart[(i - 1) / 4] + 1]]];
         If[Mod[i, 4] == 0,
          Functions[[i]] = WReshape[[IntegerPart[(i - 1) / 4] + 1]]];
        ]

In[ ]:= FunctionsReshape = ArrayReshape[Functions, {nx ny, 4}];

In[ ]:= xFunction = x;

In[ ]:= For[i = 1, i < ny, i++,
         xFunction = Join[xFunction, x]]

In[ ]:= yFunction = Table[0, {i, 1, nx ny}];

In[ ]:= For[i = 1, i ≤ nx ny, i++,
         yFunction[[i]] = y[[IntegerPart[(i - 1) / nx] + 1]]]
```

In[ ]:=
```
FunctionsReshapeFinal = Transpose[{xFunction, yFunction, FunctionsReshape[[All, 1]],
    FunctionsReshape[[All, 2]], FunctionsReshape[[All, 3]], FunctionsReshape[[All, 4]]}];
```

Export to files

Setting the directory to where we export the files

```
SetDirectory[NotebookDirectory[]]
```

SetDirectory: File specification /BackUp /home/jorge/MEGA/Tese/Notes/Tutorial_CADSOL_Kerr/Workshop_COGWDL/Kerr/ is not a string of one or more characters.

Out[ ]=
```
SetDirectory[
 /BackUp /home/jorge/MEGA/Tese/Notes/Tutorial_CADSOL_Kerr/Workshop_COGWDL/Kerr/]
```

In[ ]:=
```
Export["gridx.dat", x, "Table"]
Export["gridy.dat", y, "Table"]
```

Out[ ]=
```
gridx.dat
```

Out[ ]=
```
gridy.dat
```

In[ ]:=
```
Export["functf.dat", Functions, "Table"]
Export["funct.dat", FunctionsReshapeFinal, "Table"]
```

Out[ ]=
```
functf.dat
```

Out[ ]=
```
funct.dat
```

After doing this, remember to call the subroutine “write” when running FIDISOL/CADSOL for the first time, so the subroutine can translate the files exported from MATHEMATICA to something Fortran likes to read.