In[214]:=

```
(*==============================================================*)
(* Parameters                                                   *)
(*==============================================================*)

ClearAll[R0, A, w, ε, hFD];



R0 = SetPrecision[1.0, 50];
A  = SetPrecision[1.0*^-4, 50];
w  = SetPrecision[0.02, 50];
ε = SetPrecision[0.01, 50];

hFD = w/25;

(*==============================================================*)
(* Conformal factor ΦSafe                                       *)
(*==============================================================*)

ClearAll[ΦSafe];

ΦSafe[x_?NumericQ, y_?NumericQ, z_?NumericQ] :=
 Module[{xx = SetPrecision[x, 50],
    yy = SetPrecision[y, 50],
    zz = SetPrecision[z, 50],
    r, ϕ},

  r = Max[Sqrt[xx^2 + yy^2 + zz^2], ε];
  ϕ = -A (1 - R0/r) Exp[-((r - R0)^2)/w^2];

  SetPrecision[ϕ, 50] /. _?Negative :> -10^-30
];

(*==============================================================*)
(* Metric g_{μν} and inverse                                   *)
(*==============================================================*)

ClearAll[g, gInv];

g[x_?NumericQ, y_?NumericQ, z_?NumericQ] :=
 g[x, y, z] =
```

```
 Module[{f = Exp[2 ΦSafe[x, y, z]]},
  {{-f, 0, 0, 0},
   {0, 1/f, 0, 0},
   {0, 0, 1/f, 0},
   {0, 0, 0, 1/f}}
 ];

gInv[x_?NumericQ, y_?NumericQ, z_?NumericQ] :=
 gInv[x, y, z] = Inverse[g[x, y, z]];

(*============================================================*)
(* Finite-difference derivative                                  *)
(*============================================================*)

ClearAll[dΦ];

dΦ[{x_, y_, z_}, dir_] :=
 Module[{xx = SetPrecision[x, 50], yy = SetPrecision[y, 50],
    zz = SetPrecision[z, 50], δ = hFD},

  Switch[dir,
   1, (ΦSafe[xx + δ, yy, zz] -
     ΦSafe[xx - δ, yy, zz])/(2 δ),
   2, (ΦSafe[xx, yy + δ, zz] -
     ΦSafe[xx, yy - δ, zz])/(2 δ),
   3, (ΦSafe[xx, yy, zz + δ] -
     ΦSafe[xx, yy, zz - δ])/(2 δ)
  ]
];

(*============================================================*)
(* Einstein tensor G_{μν}                                        *)
(*============================================================*)

ClearAll[Einstein];

Einstein[x_?NumericQ, y_?NumericQ, z_?NumericQ] :=
 Module[{gradPhi},
  gradPhi = Table[dΦ[{x, y, z}, i], {i, 1, 3}];

  DiagonalMatrix[{
```

```
  -2 Total[gradPhi^2],
   2 gradPhi[[1]]^2,
   2 gradPhi[[2]]^2,
   2 gradPhi[[3]]^2
 }]
];

(*============================================================*)
(* Stress-energy tensor T_{μν}                                  *)
(*============================================================*)

ClearAll[T];
T[x_?NumericQ, y_?NumericQ, z_?NumericQ] := Einstein[x, y, z]/(8 π);

(*============================================================*)
(* Energy conditions: NEC, WEC, SEC, DEC                         *)
(*============================================================*)

nullVec = {1, 1, 0, 0};
timeVec = {1, 0, 0, 0};

ClearAll[NEC, WEC];

NEC[x_, y_, z_] := Chop[nullVec . T[x, y, z] . nullVec];
WEC[x_, y_, z_] := Chop[timeVec . T[x, y, z] . timeVec];

(*----------------------------*
   Strong Energy Condition SEC
 *----------------------------*)

ClearAll[SEC];

SEC[x_?NumericQ, y_?NumericQ, z_?NumericQ] :=
 Module[{Tval = T[x, y, z], ρ, px, py, pz},
  ρ = -Tval[[1, 1]];
  px = Tval[[2, 2]];
  py = Tval[[3, 3]];
  pz = Tval[[4, 4]];
  Chop[ρ + px + py + pz]
 ];

(*----------------------------*
```

```
   Dominant Energy Condition DEC
 *-----------------------------*)

ClearAll[DEC];

DEC[x_?NumericQ, y_?NumericQ, z_?NumericQ] :=
 Module[{Tval = T[x, y, z], ρ, px, py, pz},
  ρ  = -Tval[[1, 1]];
  px = Tval[[2, 2]];
  py = Tval[[3, 3]];
  pz = Tval[[4, 4]];

  If[ ρ ≥ Abs[px] && ρ ≥ Abs[py] && ρ ≥ Abs[pz],
    Chop[ρ],   (* DEC satisfied *)
    Chop[-ρ]  (* DEC violated *)
  ]
 ];

(*=============================================================*)
(* Exotic energy integral                                      *)
(*=============================================================*)

ClearAll[ρ, sqrtMinusDet, integrand];

ρ[r_?NumericQ] := -T[r, 0, 0][[1, 1]];
sqrtMinusDet[r_] := Exp[-2 ΦSafe[r, 0, 0]];
integrand[r_?NumericQ] := 4 π r^2 ρ[r] × sqrtMinusDet[r];

totalExoticEnergy =
 NIntegrate[
  integrand[r],
  {r, ε, 3 R0},
  WorkingPrecision → 50,
  AccuracyGoal → 8,
  PrecisionGoal → 8,
  Method → {"GlobalAdaptive", "SymbolicProcessing" → 0}
 ];

Print["\n► Total exotic (negative) energy =  ", totalExoticEnergy];

(*=============================================================*)
(* Table of NEC, WEC, SEC, DEC                                 *)
```

```
(*============================================================*)

TableForm[
 Table[{r,
    NEC[r, 0, 0],
    WEC[r, 0, 0],
    SEC[r, 0, 0],
    DEC[r, 0, 0]},
   {r, {0.02, 0.05, 0.1, 0.5, 1.0, 2.0}}],
 TableHeadings → {
  None,
  {"r", "NEC", "WEC", "SEC", "DEC"}
 }
]
```

► Total exotic (negative) energy =
 $5.5791379993861999654135926884034092186986712347022 \times 10^{-13}$

Out[246]//TableForm=

r	NEC	WEC	SEC
0.02	0	0	0
0.05	0	0	0
0.1	0	0	0
0.5	0	0	0
1.	0	$-1.9862575067911444134914986249339248696383949411160 \times 10^{-10}$	3.9725150135
2.	0	0	0