In[•]:=
```
(* ================================================*)
(*Wormhole numeric engine-Wolfram Mathematica*)(*10 mm throat=0.01 m*)
(* ================================================*)
ClearAll[Phi, dPhi, d2Phi, rreg, R0, A, w, eps, omegaBD, Lint, bBI, E0, sigmaE,
  h, metric4, invMetric4, partialMetric, christoffel4, partialGamma, riemann4,
  ricci4, scalarR4, einstein4, Ttotal4, BBD, Treq, nullK, NEC, Eprof, BIrho, BIpr,
  BIpt, TBI4, shellRho, shellP, Tshell4, Tscalar4, radialScan, radialScanLite];

(* ====================*)
(*1. PARAMETERS*)
(* ====================*)

R0 = 2.0;  (*2 m in meters*)
A = 0.01;
w = 0.2;
eps = 0.04;
omegaBD = 100.0;
Lint = 1.0;

bBI = 0.5;  (*Born-Infeld scale*)
E0 = 0.2;  (*amplitude of E(r)*)
sigmaE = 5.0;  (*width of E(r)*)

h = 1.*^-3; (*step for finite differences*)

(* ====================*)
(*2. Phi(r) and helpers*)
(* ====================*)

rreg[r_] := Sqrt[r^2 + eps^2];

Phi[r_] := Module[{rr = rreg[r]}, -A (1 - R0 / rr) Exp[-(rr - R0)^2 / w^2]];

dPhi[r_] := (Phi[r + h] - Phi[r - h]) / (2 h);
d2Phi[r_] := (Phi[r + h] - 2 Phi[r] + Phi[r - h]) / h^2;

(* ====================*)
(*3. 4D metric g_{μν}*)
(* ====================*)

(*мемоизация на метриката и инверзната*)
metric4[x : {t_, r_, th_, ph_}] :=
  metric4[x] = Module[{Ph = Phi[r], e2P, em2P, g}, e2P = Exp[2 Ph];
    em2P = Exp[-2 Ph];
    g = ConstantArray[0., {4, 4}];
    g[[1, 1]] = -e2P;
```

```
    g[[2, 2]] = em2P;
    g[[3, 3]] = r^2;
    g[[4, 4]] = r^2 Sin[th]^2;
    g];

invMetric4[x : {_, _, _, _}] := invMetric4[x] = Inverse[metric4[x]];

(* ======================*)
(*4. ∂g_{μν}/∂x^α*)
(* ======================*)

partialMetric[x : {_, _, _, _}, alpha_, mu_, nu_] :=
  partialMetric[x, alpha, mu, nu] = Module[{xp = x, xm = x}, xp[[alpha]] += h;
    xm[[alpha]] -= h;
    (metric4[xp][[mu, nu]] - metric4[xm][[mu, nu]]) / (2 h)];

(* ======================*)
(*5. Christoffel Γ^μ_{νρ}*)
(* ======================*)

christoffel4[x : {_, _, _, _}] :=
  christoffel4[x] = Module[{g = metric4[x], ginv, G}, ginv = Inverse[g];
    G = ConstantArray[0., {4, 4, 4}];
    Do[Do[Do[G[[mu, nu, ro]] = 1 / 2
          Sum[ginv[[mu, sig]] (partialMetric[x, ro, sig, nu] + partialMetric[x, nu, sig, ro] -
              partialMetric[x, sig, nu, ro]), {sig, 4}], {ro, 4}], {nu, 4}], {mu, 4}];
    G];

(* ======================*)
(*6. ∂Γ/∂x*)
(* ======================*)

partialGamma[x : {_, _, _, _}, alpha_, mu_, nu_, ro_] :=
  partialGamma[x, alpha, mu, nu, ro] = Module[{xp = x, xm = x}, xp[[alpha]] += h;
    xm[[alpha]] -= h;
    (christoffel4[xp][[mu, nu, ro]] - christoffel4[xm][[mu, nu, ro]]) / (2 h)];

(* ======================*)
(*7. Riemann R^μ_{νρσ}*)
(* ======================*)

riemann4[x : {_, _, _, _}] :=
  riemann4[x] = Module[{G = christoffel4[x], R}, R = ConstantArray[0., {4, 4, 4, 4}];
    Do[
     Do[Do[Do[R[[mu, nu, ro, sg]] = partialGamma[x, ro, mu, nu, sg] - partialGamma[x, sg, mu, nu,
            ro] + Sum[G[[mu, ro, la]] × G[[la, nu, sg]] - G[[mu, sg, la]] × G[[la, nu, ro]], {la, 4}],
        {sg, 4}], {ro, 4}], {nu, 4}], {mu, 4}];
```

```
    R];

(* ====================*)
(*8. Ricci,scalar,G*)
(* ====================*)

ricci4[x : {_, _, _, _}] :=
  ricci4[x] = Module[{Riem = riemann4[x], Ric}, Ric = ConstantArray[0., {4, 4}];
    Do[Do[Ric[[nu, sg]] = Sum[Riem[[mu, nu, mu, sg]], {mu, 4}], {sg, 4}], {nu, 4}];
    Ric];

scalarR4[x : {_, _, _, _}] := scalarR4[x] = Total[Flatten[invMetric4[x].ricci4[x]]];

einstein4[x : {_, _, _, _}] :=
  einstein4[x] = Module[{g = metric4[x], Ric = ricci4[x], R = scalarR4[x]}, Ric - 1 / 2 g R];

(* ====================*)
(*9. Total T_{μν}*)
(* ====================*)

Ttotal4[x : {_, _, _, _}] := Ttotal4[x] = einstein4[x] / (8. Pi);

(* ====================*)
(*10. Brans-Dicke*)
(* ====================*)

BBD[r_] := d2Phi[r] + 2 dPhi[r] / r;

Treq[r_] := ((3 + 2 omegaBD) / (8. Pi)) BBD[r];

(* ====================*)
(*11. Null vector,NEC*)
(* ====================*)

nullK[{t_, r_, th_, ph_}] := Module[{Ph = Phi[r]}, {Exp[-Ph], Exp[Ph], 0., 0.}];

NEC[T_, x_] := Module[{k = nullK[x]}, k.T.k];

(* ====================*)
(*12. Born-Infeld sector*)
(* ====================*)

Eprof[r_] := E0 Exp[-(r - R0)^2 / sigmaE^2];

BIrho[r_] := Module[{E = Eprof[r], s}, s = Sqrt[1 + (E^2 / bBI^2)];
   bBI^2 (s - 1)];
```

```
BIpr[r_] := -BIrho[r];
BIpt[r_] := BIrho[r];

TBI4[x : {_, r_, _, _}] :=
  Module[{rho = BIrho[r]}, DiagonalMatrix[{-rho, BIpr[r], BIpt[r], BIpt[r]}]];

(* ======================*)
(*13. Shell sector*)
(* ======================*)

shellRho[r_] := Abs[Treq[r]];
shellP[r_] := (Treq[r] + shellRho[r]) / 3.;

Tshell4[x : {_, r_, _, _}] :=
  Module[{rho = shellRho[r], p = shellP[r]}, DiagonalMatrix[{-rho, p, p, p}]];

(* ======================*)
(*14. Scalar sector*)
(* ======================*)

Tscalar4[x : {_, _, _, _}] := Tscalar4[x] = Ttotal4[x] - TBI4[x] - Tshell4[x];

(* ======================*)
(*15. Radial scans*)
(* ======================*)

(*"Лайт" скан-без Ttotal/Einstein,почти моментален*)
radialScanLite[n_Integer : 11] :=
  Module[{rvals, PhiV, BBDv, Treqv, NECbi, NECsh}, rvals = Subdivide[0.5 R0, 2. R0, n - 1];
   PhiV = Phi /@ rvals;
   BBDv = BBD /@ rvals;
   Treqv = Treq /@ rvals;
   NECbi = Table[NEC[TBI4[{0., r, Pi / 2, 0.}], {0., r, Pi / 2, 0.}], {r, rvals}];
   NECsh = Table[NEC[Tshell4[{0., r, Pi / 2, 0.}], {0., r, Pi / 2, 0.}], {r, rvals}];
   <|"r" → rvals, "Phi" → PhiV, "BBD" → BBDv,
    "Treq" → Treqv, "NEC_BI" → NECbi, "NEC_shell" → NECsh|>];

(*Пълен скан-включва Ttotal и Tscalar;по-тежък*)
radialScan[n_Integer : 3] :=
  Module[{rvals, PhiV, BBDv, Treqv, NECtot, NECbi, NECsh, NECphi},
   rvals = Subdivide[0.5 R0, 2. R0, n - 1];
   PhiV = Phi /@ rvals;
   BBDv = BBD /@ rvals;
   Treqv = Treq /@ rvals;
   NECtot = Table[NEC[Ttotal4[{0., r, Pi / 2, 0.}], {0., r, Pi / 2, 0.}], {r, rvals}];
   NECbi = Table[NEC[TBI4[{0., r, Pi / 2, 0.}], {0., r, Pi / 2, 0.}], {r, rvals}];
   NECsh = Table[NEC[Tshell4[{0., r, Pi / 2, 0.}], {0., r, Pi / 2, 0.}], {r, rvals}];
```

```
    NECphi = Table[NEC[Tscalar4[{0., r, Pi / 2, 0.}], {0., r, Pi / 2, 0.}], {r, rvals}];
    <|"r" → rvals, "Phi" → PhiV, "BBD" → BBDv, "Treq" → Treqv,
     "NEC_tot" → NECtot, "NEC_BI" → NECbi, "NEC_sh" → NECsh, "NEC_phi" → NECphi|>];

(* ======================*)
(*16. "Main" тест*)
(* ======================*)

xThroat = {0., R0, Pi / 2, 0.};

Print["--- Throat (r = R0 = ", R0, ") ---"];
Print["Phi(R0)   = ", Phi[R0]];
Print["dPhi(R0)  = ", dPhi[R0]];
Print["d2Phi(R0) = ", d2Phi[R0]];
Print["BBD(R0)   = ", BBD[R0]];
Print["Treq(R0)  = ", Treq[R0]];

Print["\nMetric gμν at throat:"];
MatrixForm[metric4[xThroat]] // Print;

Print["\nEinstein tensor Gμν at throat:"];
MatrixForm[einstein4[xThroat]] // Print;

Print["\nTotal Tμν at throat:"];
MatrixForm[Ttotal4[xThroat]] // Print;

Print["\nNEC total at throat = ", NEC[Ttotal4[xThroat], xThroat]];

(*Лек скан*)
dataLite = radialScanLite[11];
Print["\nLite radial scan keys: ", Keys[dataLite]];

(*Пълен скан с малко точки,за да не виси*)
dataFull = radialScan[3];
Print["\nFull radial scan (3 points) done."];
```

```
--- Throat (r = R0 = 2.) ---
Phi(R0)   = -1.99939×10^-6
dPhi(R0)  = -0.00499682
d2Phi(R0) = 0.0052936
BBD(R0)   = 0.000296785
Treq(R0)  = 0.00239717

Metric gμν at throat:
```

$$\begin{pmatrix} -0.999996 & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. \\ 0. & 0. & 4. & 0. \\ 0. & 0. & 0. & 4. \end{pmatrix}$$

Einstein tensor $G_{\mu\nu}$ at throat:

$$\begin{pmatrix} 0.00499763 & 0. & 0. & 0. \\ 0. & -0.00499755 & 0. & 0. \\ 0. & 0. & 0.00138554 & 0. \\ 0. & 0. & 0. & 0.00138454 \end{pmatrix}$$

Total $T_{\mu\nu}$ at throat:

$$\begin{pmatrix} 0.000198849 & 0. & 0. & 0. \\ 0. & -0.000198846 & 0. & 0. \\ 0. & 0. & 0.0000551291 & 0. \\ 0. & 0. & 0. & 0.0000550893 \end{pmatrix}$$

NEC total at throat = 4.6795×10^{-9}

Lite radial scan keys: {r, Phi, BBD, Treq, NEC_BI, NEC_shell}

Full radial scan (3 points) done.