In[ ]:=
```
Quit
```

This is the second of three notebooks which accompany the paper “Gravitational Waves on Kerr Black Holes II: Metric Reconstruction with Cosmological Constant” https://arxiv.org/abs/2510.07712. This notebook computes the linearized Einstein equation and checks that the metric perturbations defined in Notebook 1 are solutions. This notebook uses the SpinWeightedSpheroidalHarmonics package from the BHPToolkit, which can be installed from here: https://bhptoolkit.org/SpinWeightedSpheroidalHarmonics/.

1. Setup and Definitions

We call the SpinWeightedSpheroidalHarmonics package, allow subscripts on our variables, and set the working directory to this notebook’s path.

In[ ]:=
```
<< Notation`
Symbolize[ParsedBoxWrapper[SubscriptBox["_", "_"]]]
<< SpinWeightedSpheroidalHarmonics`
SetDirectory@NotebookDirectory[];
```

In[∘]:=
```
coord = {t, r, θ, ϕ};
Δ[r_] := (r)^2 - 2 M * r + a^2 - Λ/3 * (r)^2 * ((r)^2 + a^2)
Δθ = 1 + Λ/3 * a^2 * Cos[θ]^2;
Ξ = 1 + Λ/3 * a^2;
Σ = r^2 + a^2 * Cos[θ]^2;
ξ = r - ⅈ * a * Cos[θ];
ξbar = r + ⅈ * a * Cos[θ];
ωsub = {Conjugate[ω] → ωbar, Conjugate[ωbar] → ω};
K[ω_, m_] := (r^2 + a^2) * ω - a * m;
Kbar[ω_, m_] := (r^2 + a^2) * Conjugate[ω] - a * m;
Q[ω_, m_] := -a * ω * Sin[θ] + m * Csc[θ];
Qbar[ω_, m_] := -a * Conjugate[ω] * Sin[θ] + m * Csc[θ];
gBL = ϵg * {{-(Δ[r] - Δθ * a^2 Sin[θ]^2)/(Ξ^2 * Σ), 0, 0, -(a * Sin[θ]^2)/(Ξ^2 * Σ) (-Δ[r] + Δθ * (r^2 + a^2))},
    {0, Σ/Δ[r], 0, 0}, {0, 0, Σ/Δθ, 0}, {-(a * Sin[θ]^2)/(Ξ^2 * Σ) (-Δ[r] + Δθ * (r^2 + a^2)),
     0, 0, ((Δθ * ((r)^2 + a^2)^2 - Δ[r] * a^2 * Sin[θ]^2)/(Σ * Ξ^2)) * Sin[θ]^2}};
lupBL = {Ξ * (r^2 + a^2)/Δ[r], 1, 0, (a * Ξ)/Δ[r]};
nupBL = 1/(2 * Σ) * {Ξ * (r^2 + a^2), -Δ[r], 0, a * Ξ};
mupBL = 1/(Sqrt[2 Δθ] * ξbar) * {I * a * Ξ * Sin[θ], 0, Δθ, I * Ξ / Sin[θ]};
mbarupBL = 1/(Sqrt[2 Δθ] * ξ) * {-I * a * Ξ * Sin[θ], 0, Δθ, -I * Ξ / Sin[θ]};
ldownBL = gBL.lupBL // Simplify;
ndownBL = gBL.nupBL // Simplify;
mdownBL = gBL.mupBL // Simplify;
mbardownBL = gBL.mbarupBL // Simplify;
```

In[ ]:=
```
lldownBL = Simplify[Outer[Times, ldownBL, ldownBL]] /. εg^2 → 1;
lndownBL = Simplify[Outer[Times, ldownBL, ndownBL]] /. εg^2 → 1;
lmdownBL = Simplify[Outer[Times, ldownBL, mdownBL]] /. εg^2 → 1;
lmbardownBL = Simplify[Outer[Times, ldownBL, mbardownBL]] /. εg^2 → 1;
nldownBL = Simplify[Outer[Times, ndownBL, ldownBL]] /. εg^2 → 1;
nndownBL = Simplify[Outer[Times, ndownBL, ndownBL]] /. εg^2 → 1;
nmdownBL = Simplify[Outer[Times, ndownBL, mdownBL]] /. εg^2 → 1;
nmbardownBL = Simplify[Outer[Times, ndownBL, mbardownBL]] /. εg^2 → 1;
mldownBL = Simplify[Outer[Times, mdownBL, ldownBL]] /. εg^2 → 1;
mndownBL = Simplify[Outer[Times, mdownBL, ndownBL]] /. εg^2 → 1;
mmdownBL = Simplify[Outer[Times, mdownBL, mdownBL]] /. εg^2 → 1;
mmbardownBL = Simplify[Outer[Times, mdownBL, mbardownBL]] /. εg^2 → 1;
mbarldownBL = Simplify[Outer[Times, mbardownBL, ldownBL]] /. εg^2 → 1;
mbarndownBL = Simplify[Outer[Times, mbardownBL, ndownBL]] /. εg^2 → 1;
mbarmdownBL = Simplify[Outer[Times, mbardownBL, mdownBL]] /. εg^2 → 1;
mbarmbardownBL = Simplify[Outer[Times, mbardownBL, mbardownBL]] /. εg^2 → 1;
```

In[ ]:=
```
CoordProjBL = {{nndownBL, nldownBL, -nmbardownBL, -nmdownBL},
    {lndownBL, lldownBL, -lmbardownBL, -lmdownBL},
    {-mbarndownBL, -mbarldownBL, mbarmbardownBL, mbarmdownBL},
    {-mndownBL, -mldownBL, mmbardownBL, mmdownBL}};
```

In[ ]:=
```
{Cos[θ] → x, Sin[θ] → Sqrt[1 - x^2], Csc[θ] → 1/Sqrt[1 - x^2],
   Sec[θ] → 1/x, Cot[θ] → x/Sqrt[1 - x^2], Tan[θ] → Sqrt[1 - x^2]/x};
Table[Sin[k*θ] → Simplify[TrigExpand[Sin[k*θ]] /. %], {k, 1, 15}];
Table[Cos[k*θ] → Simplify[TrigExpand[Cos[k*θ]] /. %%], {k, 1, 15}];
trigsub = Join[%%%, %%, %] // Flatten;
trigunsub = {Sqrt[6 - 6*x^2] → Sqrt[6]*Sin[θ], (1 - x^2)^(3/2) → Sin[θ]^3,
   (-1 + x^2) → -Sin[θ]^2, (1 - x^2) → Sin[θ]^2, Sqrt[1 - x^2] → Sin[θ], x → Cos[θ]};
```

2. Linearized Einstein Equation

Here, we compute the linearized Einstein equation, $E_{\mu\nu} = G^{(1)}_{\mu\nu} + \epsilon_g \Lambda\, h_{\mu\nu}$, in Boyer-Lindquist coordinates in terms of the metric perturbation’s tetrad components $h_{a\,b}$. We then project this onto the tetrad vectors to compute its tetrad components.

We first express a generic metric perturbation in Boyer-Lindquist coordinates in terms of its tetrad components $\left(h_{\mu\nu} = e^a_\mu\, e^b_\nu\, h_{a\,b}\right)$.

In[ ]:=
```
htetradf = {{hllf[t, r, θ, ϕ], hlnf[t, r, θ, ϕ], hlmf[t, r, θ, ϕ], hlmbarf[t, r, θ, ϕ]},
    {hlnf[t, r, θ, ϕ], hnnf[t, r, θ, ϕ], hnmf[t, r, θ, ϕ], hnmbarf[t, r, θ, ϕ]},
    {hlmf[t, r, θ, ϕ], hnmf[t, r, θ, ϕ], hmmf[t, r, θ, ϕ], hmmbarf[t, r, θ, ϕ]},
    {hlmbarf[t, r, θ, ϕ], hnmbarf[t, r, θ, ϕ],
     hmmbarf[t, r, θ, ϕ], hmbarmbarf[t, r, θ, ϕ]}};
hfBL = Collect[Sum[CoordProjBL[[a1]][[b1]] * htetradf[[a1]][[b1]], {a1, 1, 4}, {b1, 1, 4}],
    Flatten[%], Simplify[# /. trigsub] /. trigunsub &];
```

We compute the inverse metric and Christoffel symbols in Boyer-Lindquist coordinates.

In[ ]:=
```
starttime = AbsoluteTime[];
ginvBL = (Inverse[gBL] /. trigsub // Simplify) /. trigunsub;
ΓBL =
  (Table[Sum[1/2 * ginvBL[[i]][[l]] * (D[gBL[[l]][[j]], coord[[k]]] + D[gBL[[l]][[k]], coord[[j]]] -
            D[gBL[[j]][[k]], coord[[l]]]) /. trigsub, {l, 1, 4}],
      {i, 1, 4}, {j, 1, 4}, {k, 1, 4}] // Simplify) /. trigunsub;
AbsoluteTime[] - starttime
```

Out[ ]=
```
0.303147
```

We define atomic terms to assist in simplification.

In[ ]:=
```
htetradfList = Flatten[htetradf];
terms = htetradfList;
Do[terms = Join[terms, D[htetradfList, x]], {x, coord}];
Do[terms = Join[terms, D[htetradfList, {x, 2}]], {x, coord}];
Do[Do[terms = Join[terms, D[D[htetradfList, x], y]], {y, coord}], {x, coord}];
terms = DeleteDuplicates[terms];
```

We compute the first covariant derivative of the metric perturbation $\left(\nabla_\mu h_{\nu\rho} = \partial_\mu h_{\nu\rho} - \Gamma^\sigma_{\mu\nu}\, h_{\sigma\rho} - \Gamma^\sigma_{\mu\rho}\, h_{\nu\sigma}\right)$.

In[ ]:=
```
starttime = AbsoluteTime[];
t1 = Table[D[hfBL, coord[[i]]], {i, 1, 4}];
t2 = Table[Sum[ΓBL[[k]][[i1]][[i2]] * hfBL[[k]][[i3]], {k, 1, 4}],
    {i1, 1, 4}, {i2, 1, 4}, {i3, 1, 4}];
t3 = Table[Sum[ΓBL[[k]][[i1]][[i3]] * hfBL[[i2]][[k]], {k, 1, 4}],
    {i1, 1, 4}, {i2, 1, 4}, {i3, 1, 4}];
covdhfBL = ParallelMap[
     (Collect[#, terms, (Simplify[# /. trigsub]) &]) &, t1 - t2 - t3] /. trigunsub;
AbsoluteTime[] - starttime
```

Out[ ]=
```
7.334228
```

We compute the second covariant derivative of the metric perturbation
$\left(\nabla_\mu \nabla_\nu h_{\rho\sigma} = \partial_\mu\,(\nabla_\nu h_{\rho\sigma}) - \Gamma^\tau_{\mu\nu}\,\nabla_\tau h_{\rho\sigma} - \Gamma^\tau_{\mu\rho}\,\nabla_\nu h_{\tau\sigma} - \Gamma^\tau_{\mu\sigma}\,\nabla_\nu h_{\rho\tau}\right)$. This block may take a few minutes to

complete.

In[ ]:=
```
starttime = AbsoluteTime[];
t1 = Table[D[covdhfBL, coord〚i〛], {i, 1, 4}];
t2 = Table[Sum[ΓBL〚k〛〚i1〛〚i2〛 * covdhfBL〚k〛〚i3〛〚i4〛, {k, 1, 4}],
    {i1, 1, 4}, {i2, 1, 4}, {i3, 1, 4}, {i4, 1, 4}];
t3 = Table[Sum[ΓBL〚k〛〚i1〛〚i3〛 * covdhfBL〚i2〛〚k〛〚i4〛, {k, 1, 4}],
    {i1, 1, 4}, {i2, 1, 4}, {i3, 1, 4}, {i4, 1, 4}];
t4 = Table[Sum[ΓBL〚k〛〚i1〛〚i4〛 * covdhfBL〚i2〛〚i3〛〚k〛, {k, 1, 4}],
    {i1, 1, 4}, {i2, 1, 4}, {i3, 1, 4}, {i4, 1, 4}];
covdcovdhfBL = ParallelMap[
     (Collect[#, terms, (Simplify[# /. trigsub]) &]) &, t1 - t2 - t3 - t4] //. trigunsub;
AbsoluteTime[] - starttime
```

Out[ ]= 250.863442

We now compute various contractions of the second covariant derivative of the metric perturbation and then combine these terms to yield the linearized Einstein equation $G^{(1)}_{\mu\nu} + \epsilon_g \Lambda h_{\mu\nu}$. Each of these blocks may take a minute or two to run.

Term 1: $\nabla^2 h_{\mu\nu}$

In[ ]:=
```
starttime = AbsoluteTime[];
ParallelTable[Sum[ginvBL〚k〛〚l〛 * covdcovdhfBL〚k〛〚l〛〚i1〛〚i2〛, {k, 1, 4}, {l, 1, 4}],
   {i1, 1, 4}, {i2, 1, 4}];
t1 = ParallelMap[(Collect[#, terms, (Simplify[# /. trigsub]) &]) &, %] //. trigunsub;
AbsoluteTime[] - starttime
```

Out[ ]= 52.909047

Term 2: $\nabla^\rho \nabla_\mu h_{\nu\rho} + \nabla^\rho \nabla_\nu h_{\mu\rho}$

In[ ]:=
```
starttime = AbsoluteTime[];
ParallelTable[Sum[ginvBL〚k〛〚l〛 * covdcovdhfBL〚k〛〚i1〛〚i2〛〚l〛, {k, 1, 4}, {l, 1, 4}],
   {i1, 1, 4}, {i2, 1, 4}];
t2 = ParallelMap[(Collect[#, terms, (Simplify[# /. trigsub]) &]) &,
     % + Transpose[%]] //. trigunsub;
AbsoluteTime[] - starttime
```

Out[ ]= 61.026956

Term 3: $\nabla_\mu \nabla_\nu h$

In[ ]:=
```
starttime = AbsoluteTime[];
ParallelTable[Sum[ginvBL〚k〛〚l〛 * covdcovdhfBL〚i1〛〚i2〛〚k〛〚l〛, {k, 1, 4}, {l, 1, 4}],
   {i1, 1, 4}, {i2, 1, 4}];
t3 = ParallelMap[(Collect[#, terms, (Simplify[# /. trigsub]) &]) &, %] //. trigunsub;
AbsoluteTime[] - starttime
```

Out[ ]= 23.610457

Term 4: $\nabla^2 h$

```
starttime = AbsoluteTime[];
Sum[ginvBL[[k]][[l]] * ginvBL[[m]][[n]] * covdcovdhfBL[[k]][[l]][[m]][[n]],
  {k, 1, 4}, {l, 1, 4}, {m, 1, 4}, {n, 1, 4}];
t4 = Collect[%, terms, Simplify[# /. trigsub] &] //. trigunsub;
AbsoluteTime[] - starttime
```

Out[ ]= 5.800067

Term 5: $\nabla^\rho \nabla^\sigma h_{\rho\sigma}$

```
In[ ]:= starttime = AbsoluteTime[];
Sum[ginvBL[[k]][[l]] * ginvBL[[m]][[n]] * covdcovdhfBL[[l]][[n]][[k]][[m]],
  {k, 1, 4}, {l, 1, 4}, {m, 1, 4}, {n, 1, 4}];
t5 = Collect[%, terms, (Simplify[# /. trigsub]) &] //. trigunsub;
AbsoluteTime[] - starttime
```

Out[ ]= 6.446038

All Terms Together:

$$G^{(1)}_{\mu\nu} + \epsilon_g \Lambda h_{\mu\nu} = -\nabla^2 h_{\mu\nu} + \nabla^\rho \nabla_\mu h_{\nu\rho} + \nabla^\rho \nabla_\nu h_{\mu\rho} - \nabla_\mu \nabla_\nu h + g_{\mu\nu} \left(\nabla^2 h - \nabla^\rho \nabla^\sigma h_{\rho\sigma}\right) + \epsilon_g \Lambda (g_{\mu\nu} h - 2 h_{\mu\nu})$$

```
In[ ]:= starttime = AbsoluteTime[];
-t1 + t2 - t3 + gBL * (t4 - t5) +
  ϵg * Λ * (gBL * Sum[ginvBL[[i]][[j]] * hfBL[[i]][[j]], {i, 1, 4}, {j, 1, 4}] - 2 hfBL);
LinEinsteinBL =
  ParallelMap[(Collect[#, terms, Simplify[# /. trigsub] //. {ϵg^2 → 1, ϵg^-2 → 1} &]) &,
    %] //. trigunsub;
AbsoluteTime[] - starttime
```

Out[ ]= 40.127076

We now compute the tetrad components E_{ab} by projecting the Boyer-Lindquist coordinate components $E_{\mu\nu}$ onto the tetrad vectors. Due to the conjugation symmetry $\overline{E}_{ab} = \overline{E_{\mu\nu}\, e_a^\mu\, e_b^\nu} = E_{\mu\nu}\, \overline{e}_a^\mu\, \overline{e}_b^\nu \equiv E_{\overline{a}\,\overline{b}}$, we only have seven independent components.

In[∘]:=
```
starttime = AbsoluteTime[];
LinEinsteinll =
  Collect[Sum[LinEinsteinBL[[i, j]] * lupBL[[i]] * lupBL[[j]], {i, 1, 4}, {j, 1, 4}],
    terms, Simplify[# /. trigsub] //. {ϵg^2 → 1, ϵg^-2 → 1} &] //. trigunsub;
LinEinsteinnn =
  Collect[Sum[LinEinsteinBL[[i, j]] * nupBL[[i]] * nupBL[[j]], {i, 1, 4}, {j, 1, 4}],
    terms, Simplify[# /. trigsub] //. {ϵg^2 → 1, ϵg^-2 → 1} &] //. trigunsub;
LinEinsteinmm =
  Collect[Sum[LinEinsteinBL[[i, j]] * mupBL[[i]] * mupBL[[j]], {i, 1, 4}, {j, 1, 4}],
    terms, Simplify[# /. trigsub] //. {ϵg^2 → 1, ϵg^-2 → 1} &] //. trigunsub;
LinEinsteinln =
  Collect[Sum[LinEinsteinBL[[i, j]] * lupBL[[i]] * nupBL[[j]], {i, 1, 4}, {j, 1, 4}],
    terms, Simplify[# /. trigsub] //. {ϵg^2 → 1, ϵg^-2 → 1} &] //. trigunsub;
LinEinsteinlm =
  Collect[Sum[LinEinsteinBL[[i, j]] * lupBL[[i]] * mupBL[[j]], {i, 1, 4}, {j, 1, 4}],
    terms, Simplify[# /. trigsub] //. {ϵg^2 → 1, ϵg^-2 → 1} &] //. trigunsub;
LinEinsteinnm =
  Collect[Sum[LinEinsteinBL[[i, j]] * nupBL[[i]] * mupBL[[j]], {i, 1, 4}, {j, 1, 4}],
    terms, Simplify[# /. trigsub] //. {ϵg^2 → 1, ϵg^-2 → 1} &] //. trigunsub;
LinEinsteinmmbar =
  Collect[Sum[LinEinsteinBL[[i, j]] * mupBL[[i]] * mbarupBL[[j]], {i, 1, 4}, {j, 1, 4}],
    terms, Simplify[# /. trigsub] //. {ϵg^2 → 1, ϵg^-2 → 1} &] //. trigunsub;
AbsoluteTime[] - starttime
```

Out[∘]=
```
24.659232
```

In[∘]:=
```
(* Save the results to expedite future notebook usage. *)
LinEinsteinab = {LinEinsteinll, LinEinsteinnn, LinEinsteinmm, LinEinsteinln,
   LinEinsteinlm, LinEinsteinlm, LinEinsteinnm, LinEinsteinmmbar};
DumpSave["./linearized-Einstein-tensor/LinEinsteinab.mx", LinEinsteinab];
```

3. Radial and Angular Mode Functions

In order to check that the metric perturbations defined above in terms of $R^{(\pm 2)}_{\omega l m}(r)$ and $S^{(\pm 2)}_{\omega l m}(\theta)$ solve linearized Einstein equations, we will need to use the radial and angular equations of motion.

We first define lists containing the radial and angular functions to be used in later simplifications.

In[ ]:=
```
{R[s, ω, l, m, r], R[s, -ωbar, l, -m, r],
  Rbar[s, ω, l, m, r], Rbar[s, -ωbar, l, -m, r]};
Table[%, {s, {2, -2}}];
Rlist = Flatten[{%, D[%, r], D[%, {r, 2}]}];
{S[s, ω, l, m, θ], S[s, -ωbar, l, -m, θ],
  Sbar[s, ω, l, m, θ], Sbar[s, -ωbar, l, -m, θ]};
Table[%, {s, {2, -2}}];
Slist = Flatten[{%, D[%, θ], D[%, {θ, 2}]}];
```

We define the radial and angular Teukolsky equations.

In[ ]:=
```
eqR = Δ[r]^-s * D[Δ[r]^(s+1) * D[R[s, ω, l, m, r], r], r] +
   ((Ξ^2 * K[ω, m]^2 - s * I * Ξ * K[ω, m] * Δ'[r])/Δ[r] + 4 I * s * ω * Ξ * r +
      s * Δ''[r] - 2 (2 s - 1) * (s - 1) * Λ/3 * r^2 + s/3 a^2 Λ) *
    R[s, ω, l, m, r] - (λ[s, ω, l, m] + 2 s) * R[s, ω, l, m, r];
eqS = 1/Sin[θ] D[Δθ * Sin[θ] * D[S[s, ω, l, m, θ], θ], θ] +
   (ω^2 * Ξ^2 (-(a^2 * Sin[θ]^2)/Δθ) + 2 * m * ω * a * Ξ^2/Δθ + m^2 * Ξ^2 (-1/Sin[θ]^2 + (a^2 * Λ)/(3 Δθ) * Cot[θ]^2) - 2 ω *
      s * a * Ξ^2 * Cos[θ]/Δθ - 2 m * s * Ξ * (2 - Ξ/Δθ) Cos[θ]/Sin[θ]^2 - 2 (2 s^2 + 1) * Λ/3 * a^2 * Cos[θ]^2 -
      (+ s^2 * Δθ * (D[Δθ, θ]/(2 Δθ) + Cot[θ])^2) + λ[s, ω, l, m] + s) * S[s, ω, l, m, θ];
eqRbar = Δ[r]^-s * D[Δ[r]^(s+1) * D[Rbar[s, ω, l, m, r], r], r] +
   ((Ξ^2 * Kbar[ω, m]^2 + s * I * Ξ * Kbar[ω, m] * Δ'[r])/Δ[r] -
      4 I * s * ωbar * Ξ * r + s * Δ''[r] - 2 (2 s - 1) * (s - 1) * Λ/3 * r^2 + s/3 a^2 Λ) *
    Rbar[s, ω, l, m, r] - (λbar[s, ω, l, m] + 2 s) * Rbar[s, ω, l, m, r];
eqSbar = 1/Sin[θ] D[Δθ * Sin[θ] * D[Sbar[s, ω, l, m, θ], θ], θ] +
   (ωbar^2 * Ξ^2 (-(a^2 * Sin[θ]^2)/Δθ) + 2 * m * ωbar * a * Ξ^2/Δθ +
      m^2 * Ξ^2 (-1/Sin[θ]^2 + (a^2 * Λ)/(3 Δθ) * Cot[θ]^2) - 2 ωbar * s * a * Ξ^2 * Cos[θ]/Δθ -
      2 m * s * Ξ * (2 - Ξ/Δθ) Cos[θ]/Sin[θ]^2 - 2 (2 s^2 + 1) * Λ/3 * a^2 * Cos[θ]^2 -
      (+ s^2 * Δθ * (D[Δθ, θ]/(2 Δθ) + Cot[θ])^2) + λbar[s, ω, l, m] + s) * Sbar[s, ω, l, m, θ];
```

Using the above equations of motion, we define rules to remove second radial and angular derivatives of the mode functions. We then check that these rules yield zero when applied to the equations of motion.

```
In[•]:= Rsub = {D[R[s0_, ω0_, l0_, m0_, r], {r, 2}] :>
        Evaluate[Simplify[D[R[s, ω, l, m, r], {r, 2}] /.
             Solve[eqR == 0, D[R[s, ω, l, m, r], {r, 2}]]][[1]] /.
         {s → s0, ω → ω0, ωbar → Conjugate[ω0], l → l0, m → m0},
       D[Rbar[s0_, ω0_, l0_, m0_, r], {r, 2}] :>
        Evaluate[Simplify[D[Rbar[s, ω, l, m, r], {r, 2}] /.
             Solve[eqRbar == 0, D[Rbar[s, ω, l, m, r], {r, 2}]]][[1]] /.
         {s → s0, ω → ω0, ωbar → Conjugate[ω0], m → m0}};
      Ssub = {D[S[s0_, ω0_, l0_, m0_, θ], {θ, 2}] :>
        Evaluate[Simplify[D[S[s, ω, l, m, θ], {θ, 2}] /.
             Solve[eqS == 0, D[S[s, ω, l, m, θ], {θ, 2}]]][[1]] /.
         {s → s0, ω → ω0, ωbar → Conjugate[ω0], l → l0, m → m0},
       D[Sbar[s0_, ω0_, l0_, m0_, θ], {θ, 2}] :>
        Evaluate[Simplify[D[Sbar[s, ω, l, m, θ], {θ, 2}] /.
             Solve[eqSbar == 0, D[Sbar[s, ω, l, m, θ], {θ, 2}]]][[1]] /.
         {s → s0, ω → ω0, ωbar → Conjugate[ω0], m → m0}};
```

```
In[•]:= {eqR == 0, eqRbar == 0} /. Rsub /. ωsub // Simplify
        {eqS == 0, eqSbar == 0} /. Ssub /. ωsub // Simplify
Out[•]= {True, True}
Out[•]= {True, True}
```

4. Checking the Linearized Einstein Equations

We import the expressions for the tetrad components of $h_{\mu\nu}$ generated by MetricPerturbations.nb.

```
In[•]:= << "metric-perturbations/habIRG.mx"
        << "metric-perturbations/habORG.mx"
In[•]:= << "linearized-Einstein-tensor/LinEinsteinab.mx"
```

4a. Checking IRG Einstein Equations

Here we check the linearized Einstein equation in IRG. We define a rule that replaces the generic functions representing the tetrad components with the imported tetrad components.

Recall that the tetrad basis has the ordering $(l^\mu, n^\mu, m^\mu, \overline{m}^\mu)$. For example, this means that `h_{nn}` is given by habIRG/ORG[[2,2]].

In[ ]:=
```
IRGsub = {hllf → Function[{t, r, θ, ϕ}, 0],
    hlnf → Function[{t, r, θ, ϕ}, 0], hlmf → Function[{t, r, θ, ϕ}, 0],
    hlmbarf → Function[{t, r, θ, ϕ}, 0], hmmbarf → Function[{t, r, θ, ϕ}, 0],
    hnnf → Function[{x1, x2, x3, x4}, habIRG⟦2, 2⟧ /. {t → x1, r → x2, θ → x3, ϕ → x4}],
    hnmf → Function[{x1, x2, x3, x4}, habIRG⟦2, 3⟧ /. {t → x1, r → x2, θ → x3, ϕ → x4}],
    hnmbarf →
     Function[{x1, x2, x3, x4}, habIRG⟦2, 4⟧ /. {t → x1, r → x2, θ → x3, ϕ → x4}],
    hmmf → Function[{x1, x2, x3, x4}, habIRG⟦3, 3⟧ /. {t → x1, r → x2, θ → x3, ϕ → x4}],
    hmbarmbarf →
     Function[{x1, x2, x3, x4}, habIRG⟦4, 4⟧ /. {t → x1, r → x2, θ → x3, ϕ → x4}]
   };
```

We then apply this rule to the seven independent components of the linearized Einstein tensor. It is substantially faster to separate the four cases where only one of A, B, $\overline{A}$, or $\overline{B}$ is non-zero and check them in parallel. (In one case, this is not necessary, as the particular G_{ab} component vanishes immediately due to the gauge condition.) After doing this separation, we apply the radial and angular rules from above, first replacing the fourth derivatives of $R^{(\pm 2)}_{\omega l m}(r)$ and $S^{(\pm 2)}_{\omega l m}(\theta)$, then the third, and finally the second. Collecting terms and simplifying will yield zero in all four cases, and since the coefficients are arbitrary, this is sufficient to show that the expressions imported above solve the linearized Einstein equations. Most of these these checks will take only a few minutes, but the (n, m) component for IRG and the (l, m) component for ORG may take about six minutes and 12 minutes to complete, respectively.

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinll /. IRGsub
AbsoluteTime[] - starttime
```

Out[ ]= 0

Out[ ]= 0.002689

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinln /. IRGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
   % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
   ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 77.607355

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinlm /. IRGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 31.079694

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinnn /. IRGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 146.384561

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinnm /. IRGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 381.676569

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinmm /. IRGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 114.102875

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinmmbar /. IRGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 85.897159

4b. Checking ORG Einstein Equations

The procedure here is exactly the same as above.

In[ ]:=
```
ORGsub = {hlnf → Function[{t, r, θ, ϕ}, 0],
   hnnf → Function[{t, r, θ, ϕ}, 0], hnmf → Function[{t, r, θ, ϕ}, 0],
   hnmbarf → Function[{t, r, θ, ϕ}, 0], hmmbarf → Function[{t, r, θ, ϕ}, 0],
   hllf → Function[{x1, x2, x3, x4}, habORG[[1, 1]] /. {t → x1, r → x2, θ → x3, ϕ → x4}],
   hlmf → Function[{x1, x2, x3, x4}, habORG[[1, 3]] /. {t → x1, r → x2, θ → x3, ϕ → x4}],
   hlmbarf →
    Function[{x1, x2, x3, x4}, habORG[[1, 4]] /. {t → x1, r → x2, θ → x3, ϕ → x4}],
   hmmf → Function[{x1, x2, x3, x4}, habORG[[3, 3]] /. {t → x1, r → x2, θ → x3, ϕ → x4}],
   hmbarmbarf →
    Function[{x1, x2, x3, x4}, habORG[[4, 4]] /. {t → x1, r → x2, θ → x3, ϕ → x4}]
  };
```

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinll /. ORGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 109.441566

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinln /. ORGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 74.799602

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinlm /. ORGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 741.164750

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinnn /. ORGsub
AbsoluteTime[] - starttime
```

Out[ ]= 0

Out[ ]= 0.015555

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinnm /. ORGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 251.999401

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinmm /. ORGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &,
  %] //. trigunsub
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 111.642101

In[ ]:=
```
starttime = AbsoluteTime[];
LinEinsteinmmbar /. ORGsub;
{% /. Abar → 0 /. B → 0 /. Bbar → 0, % /. A → 0 /. B → 0 /. Bbar → 0,
  % /. A → 0 /. Abar → 0 /. Bbar → 0, % /. A → 0 /. Abar → 0 /. B → 0};
% /. D[Rsub, {r, 2}] /. D[Rsub, r] /. Rsub /. D[Ssub, {θ, 2}] /. D[Ssub, θ] /. Ssub /.
  ωsub;
ParallelMap[(Collect[#, Rlist, Collect[#, Slist, Simplify[# /. trigsub] &] &]) &, %]
AbsoluteTime[] - starttime
```

Out[ ]= {0, 0, 0, 0}

Out[ ]= 90.334495