In[2]:=
```
ClearAll["Global`*"];
$MaxExtraPrecision = 100;
Potential[r_, l_, m_] := ((1 - 2 / r) (l + l^2 + 2 / r + m^2 r^2)) / r^2;

FindRmax[l_?NumericQ, m_?NumericQ, prec_ : 50] := Module[{rMax},
  rMax = r /. FindRoot[D[Potential[r, l, m], r] == 0, {r, 3}, WorkingPrecision → prec];
  SetPrecision[rMax, prec]];
lambda0[u_, l_, m_, ω_] := Module[{k = Sqrt[ω^2 - m^2]},
  ((-1 + 4 I ω) + u (4 - 3 u + 4 I (-2 + u) ω - 8 I k + 4 I u k)) / ((-1 + u)^2 u)];
s0[u_, l_, m_, ω_] := Module[{k = Sqrt[ω^2 - m^2]},
  (l (-1 + u) + l^2 (-1 + u) - 4 m^2 (1 + (-3 + u) u) + (-1 + u)
     (1 - 4 I k - 4 ω (I + 4 ω + 4 k) + u (-1 + 4 I k + 4 ω (I + 2 ω + 2 k)))) / ((-1 + u)^3 u)];
AIMIterateNumeric[c0_List, d0_List, Imax_Integer, Nmax_Integer] :=
 Module[{cMat, dMat, n, i, delta}, cMat = ConstantArray[0, {Nmax + 1, Imax + 2}];
  dMat = ConstantArray[0, {Nmax + 1, Imax + 2}];
  cMat[[1, 1 ;; Imax + 1]] = c0;
  dMat[[1, 1 ;; Imax + 1]] = d0;
  For[n = 1, n ≤ Nmax, n++, For[i = 0, i ≤ Imax, i++, cMat[[n + 1, i + 1]] =
     (i + 1) cMat[[n, i + 2]] + dMat[[n, i + 1]] + Sum[c0[[k + 1]] × cMat[[n, i - k + 1]], {k, 0, i}];
    dMat[[n + 1, i + 1]] = (i + 1) dMat[[n, i + 2]] +
      Sum[d0[[k + 1]] × cMat[[n, i - k + 1]], {k, 0, i}];];];
  delta = dMat[[Nmax + 1, 1]] × cMat[[Nmax, 1]] - dMat[[Nmax, 1]] × cMat[[Nmax + 1, 1]]; delta];
(*Delta AIM*)
DeltaNumeric[l_?NumericQ, m_?NumericQ, ω_?NumericQ, uMax_, Imax_Integer,
  Nmax_Integer, prec_ : 50] := Module[{seriesLambda, seriesS, c0, d0},
  seriesLambda =
   Collect[Normal@Series[lambda0[u, l, m, ω], {u, uMax, Imax}], u - uMax];
  seriesS = Collect[Normal@Series[s0[u, l, m, ω], {u, uMax, Imax}], u - uMax];
  c0 = Table[Coefficient[seriesLambda, u - uMax, i], {i, 0, Imax}];
  d0 = Table[Coefficient[seriesS, u - uMax, i], {i, 0, Imax}];
  SetPrecision[AIMIterateNumeric[
```

```
    SetPrecision[c0, prec], SetPrecision[d0, prec], Imax, Nmax], prec]];
FindQNM[l_?NumericQ, m_?NumericQ, ωGuess_?NumericQ,
  Imax_ : 25, Nmax_ : 25, prec_ : 80] := Module[{rMax, uMax},
  rMax = FindRmax[l, m, prec];
  uMax = SetPrecision[1 - 2 / rMax, prec];
  FindRoot[DeltaNumeric[l, m, ω, uMax, Imax, Nmax, prec] == 0,
   {ω, ωGuess}, WorkingPrecision → prec, AccuracyGoal → Floor[prec / 2],
   PrecisionGoal → Floor[prec / 2], MaxIterations → 200]]
```

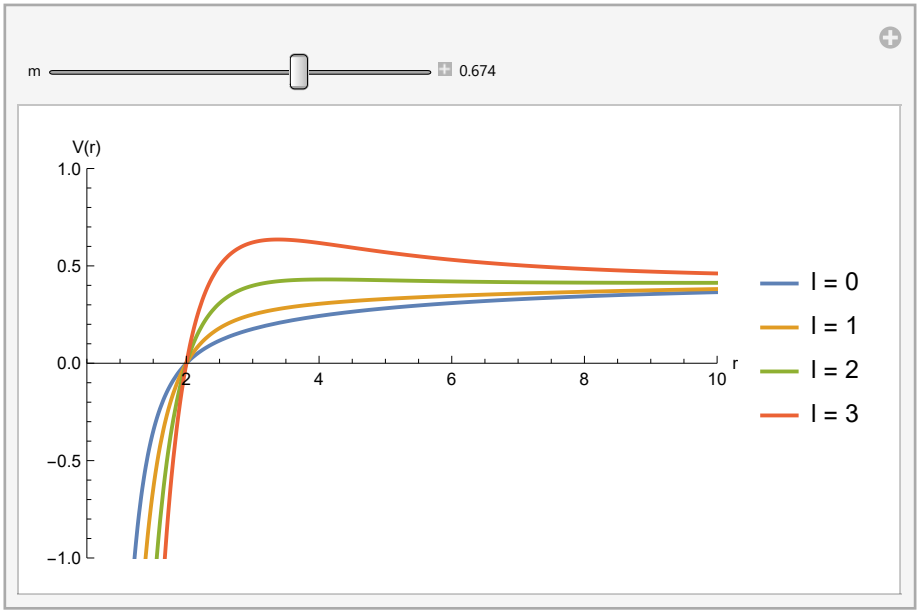
In[ ]:=
```
{time, sol} = AbsoluteTiming[FindQNM[0, 0, 11 / 100 - 10 / 100 I, 60, 60, 60]]
```

Out[ ]=
```
{17.9592, {ω → 0.11045438221818001165927597904916077288057545945372631593178 9 -
    0.10489453350368925591407231514941976723867677380493633619066 2 i}}
```

In[ ]:=
```
(*https://arxiv.org/pdf/2206.03512 Comparar valores*)
```

In[11]:=
```
lvals = {0, 1, 2, 3};
Manipulate[Plot[Evaluate[Table[Potential[r, l, m], {l, lvals}]],
  {r, 0.5, 10}, PlotRange → {{0.5, 10}, {-1, 1}}, PlotStyle → Thick,
  AxesLabel → {"r", "V(r)"}, PlotLegends → ("l = " <> ToString[#] & /@ lvals)],
 {{m, 0.1, "m"}, 0, 1, Appearance → "Labeled"}]
```

Out[11]=



```
(*Guess con WKB 1°*)
QNMWKB[l_?NumericQ, m_?NumericQ, n_Integer:0, prec_:50] :=
 Module[{r0, pot0, d2pot0, omega2, omega}, r0 = FindRmax[l, m, prec];
  pot0 = Potential[r0, l, m];
  d2pot0 = (1 - 2 / r0) * D[(1 - 2 / r) * D[Potential[r, l, m], r], r] /. r → r0;
  omega2 = pot0 - I (n + 1 / 2) Sqrt[-2 d2pot0];
  omega = Sqrt[omega2];
  N[omega, prec]]

In[ ]:= FindQNM1[l_?NumericQ, m_?NumericQ, n_Integer, Imax_:25, Nmax_:25, prec_:80] :=
 Module[{rMax, uMax, omegaGuess}, rMax = FindRmax[l, m, prec];
  uMax = SetPrecision[1 - 2 / rMax, prec];
  omegaGuess = QNMWKB[l, m, n, prec];
  FindRoot[DeltaNumeric[l, m, ω, uMax, Imax, Nmax, prec] == 0,
   {ω, omegaGuess}, WorkingPrecision → prec, AccuracyGoal → Floor[prec / 2],
   PrecisionGoal → Floor[prec / 2], MaxIterations → 200]]

In[ ]:= {time, sol} = AbsoluteTiming[FindQNM1[2, 0, 2, 25, 25, 50]]

Out[ ]= {2.90512, {ω → 0.43054416861973394721722579079079882957877931968543 -
     0.50855581055702274054065695543086913367532692171612 i}}
```