```
(* Matrix dimensions *)
numCoefficents := 10;

numInterPoints := 20;

d[m_, n_] := KroneckerDelta[m, n];

lP[n_, x_] := LegendreP[n, x];

mSum[n_, x_] := (-n (n + 1) lP[n, x] + p^2 x^2 lP[n, x]);
```

$$\texttt{mSumExpanded[m\_, n\_] :=} \left(\frac{-2\,n\,(n+1)}{2\,n+1} + p^2\,\frac{2\,(2\,n^2+2\,n-1)}{(2\,n+3)\,(2\,n+1)\,(2\,n-1)}\right) d[m, n] + \left(p^2\,\frac{2\,(n+1)\,(n+2)}{(2\,n+1)\,(2\,n+3)\,(2\,n+5)}\right) d[m, n+2] + \left(p^2\,\frac{2\,n\,(n-1)}{(2\,n+1)\,(2\,n-1)\,(2\,n-3)}\right) d[m, n-2];$$

```
lSumExpanded[m_, n_] :=
  (-2 p n (2 n + 1) - (2 p - 1) n^2 + (-2 p + 2 R) (2 n + 1) + (-4 p + 1) n + (-2 p - p^2 + 2 R))
    d[m, n] + (2 p n (n + 1) - (-2 p + 2 R) (n + 1)) d[m, n + 1] +
   (2 p n^2 + (2 p - 1) n (2 n - 1) - (-2 p + 2 R) n + (4 p - 3) n) d[m, n - 1] +
   (-(2 p - 1) n (n - 1)) d[m, n - 2];

tableEofR = {};


allRs = Table[i, {i, 0.2, 5, 0.2}];
allRs  = Join[allRs, Table[i, {i, 5.5, 9, 0.5}]];

CalcE[radius_, max_, np_] := Module[{matrixM, matrixL, ee, q,
   pe, rm, pMax, pMin, pStep, eigenM, eigenL, pgrid, mFunc, lFunc},

    (* m = rows, n = columns.
     Sums go from 0 (zero), matrix indices go from 1
   *)


  (* M matrix *)
  matrixM = Table[0, {i, 1, max}];
  Do[
   matrixM[[m + 1]] = Table[mSumExpanded[m, n] ((2 m + 1)/2), {n, 0, max - 1}],
   {m, 0, max - 1}
  ];
```

```
(* L Matrix  For Power series solution *)

matrixL = Table[0, {i, 1, max}];

Do[
 matrixL[[m + 1]] = Table[lSumExpanded[m, n], {n, 0, max - 1}],
 {m, 0, max - 1}
];

q = 2.;
pMax = (2. q radius)/2. + 1.;
pMin = radius/2.;
pStep = (pMax - pMin)/np;

(*Output *)

(* {p, eigenvalue @ p } *)
eigenM = Table[0, {i, 1, np}, {j, 1, 2}];
eigenL = Table[0, {i, 1, np}, {j, 1, 2}];
pgrid = Table[pMin + (i - 1) * pStep, {i, 1, np}];

rm[x_] := If[Im[x] ≠ 0, 10^-99, x];

Do[
 eigenM[[i]][[2]] =
  Sort[Map[rm, Eigenvalues[N[matrixM /. p → pgrid[[i]], 10.]]], Greater][[1]];
 eigenM[[i]][[1]] = pgrid[[i]];
 eigenL[[i]][[2]] = Sort[Map[rm,
     (Eigenvalues[N[matrixL /. {R → radius, p → pgrid[[i]]}, 10.]] (-1))]][[1]];
 eigenL[[i]][[1]] = pgrid[[i]],
 {i, 1, np}
];

(* Now Interpolation *)

mFunc = (eigenM // Interpolation) ;
lFunc  = (eigenL // Interpolation) ;

pe = x /. FindRoot[mFunc[x] == lFunc[x], {x, radius}];
```

```
  ee = -4 (pe/radius)^2;
  ee
 ]


(* Syntax:
    [E(R), E(R)+1/R] = CalcE[1,numCoefficents, numInterPoints]
*)

Energies := CalcE[#, numCoefficents, numInterPoints] & ;
Plot[{Energies[r], Energies[r] + 1/r}, {r, 0.1, 10},
 AxesOrigin → {0, 0}, PlotLabels → {E (R), E (R) + 1/R}]
```

Out[117]=