In[102]:=

```
Clear[m];
Clear[l];
Clear[K];
Clear[I1];
Clear[I2];
Clear[C0];
Clear[C1];
Clear[C2];
Clear[DotC];
Clear[n0];
Clear[fbare];
Clear[ft];
Clear[ξ];
Clear[fGS];
Clear[TabSingle];
Clear[TabAll];
Clear[x];
```

Functions

In[119]:=

```
p = 1;
h = 1;
Nb = 100;
I1[p1_, p2_] := Beta[p1 + 1, p2 - p1 - 1] * 2^(p1 - p2 + 1);
I2[p1_, p2_] := I1[p1, p2 + 1];
(**)
C0[n_] := 8 * n^2;
C1[n_] := 2 (1 + 2 * n * (n + 1) - 3 (2 n + 1));
C2[n_] := 2 (1 - 2 (n + 1));
(**)
K[k_, n_, l_, m_] :=
  -Which[n == 0 && m == 0, C1[n] * I1[k + n, k + n + 2 p + 2] + C2[n] * I1[k + n + 1, k + n + 2 p + 2] -
     l (l + 1) I1[k + n, k + n + 2 p], n > 0 && m == 0, C0[n] * I1[k + n - 1, k + n + 2 p + 2] +
     C1[n] * I1[k + n, k + n + 2 p + 2] + C2[n] * I1[k + n + 1, k + n + 2 p + 2] -
     l (l + 1) I1[k + n, k + n + 2 p], n > 0 && m > 0, C0[n] * I1[k + n - 1, k + n + 2 p + 2] +
     C1[n] * I1[k + n, k + n + 2 p + 2] + C2[n] * I1[k + n + 1, k + n + 2 p + 2] -
     l (l + 1) I1[k + n, k + n + 2 p] - m^2 * I2[k + n - 1, k + n + 2 p]];
DotC[k_, n_, l_, m_] :=
  Which[k == 0 && n == 0, K[0, 0, l, m] + p K[0, 1, l, m] + p K[1, 0, l, m] + p^2 K[1, 1, l, m],
   k == 0 && n > 0, K[0, n + 1, l, m] + p K[1, n + 1, l, m], k > 0 && n == 0,
   K[k + 1, 0, l, m] + p K[k + 1, 1, l, m], k > 0 && n > 0, K[k + 1, n + 1, l, m]];
ComputeY1[l_, m_, n0_] := Module[{MatC, R1, Y1}, Print["Computing Gram Matrix..."];
```

```
  MatC = ParallelTable[DotC[k, n, l, m], {k, n0, Nb}, {n, n0, Nb}];
  Print["Computing Cholesky decomposition..."];
  R1 = CholeskyDecomposition[MatC];
  Print["Computing orthogonal coefficients..."];
Y1 = Inverse[R1]; Return[Y1]]
(**)
fbare[x_, n_, l_, m_] := If[n > 0, (x)^n, 1];
ft[x_, n_, l_, m_] :=
  If[n == 0, fbare[x, 0, l, m] + p fbare[x, 1, l, m], fbare[x, n + 1, l, m]];
ξ[x_] := (1 + h x) / (1 - x);
(**)
fGS[x_, n_, l_, m_, tabY1_, n0_] := Module[{y, z}, Clear[y];
   Clear[z];
   y = ParallelSum[tabY1[[k + 1 - n0, n + 1 - n0]] * ft[x, k, l, m], {k, n0, n}];
   z = y // Simplify;
   Return[z]];
(**)
TabSingle[Nr_, l_, m_, tabY1_, n0_] := Module[{fg, nbx, tab}, Clear[fg];
   Clear[nbx];
   fg = fGS[x, Nr, l, m, tabY1, n0] / (ξ[x] + h);
   nbx = Max[Nr * 100, 100];
   tab = ParallelTable[{i / nbx, If[i < nbx, fg /. x → i / nbx, 0]}, {i, 0, nbx}] // N;
   If[Mod[Nr, 10] == n0, Print["n=", Nr]]; Return[tab]];
(**)
TabAll[Nmax_, l_, m_] := Module[{tab, Y1, n0}, n0 = If[m == 0, 0, 1];
   Y1 = ComputeY1[l, m, n0];
   Print["Sampling the orthonormal basis..."];
   tab = Table[TabSingle[Nr, l, m, Y1, n0], {Nr, n0, Nmax}];
   Return[tab]];
(**)
ExportTab[Nmax_, l_, m_] := Module[{tabAll, n0}, tabAll = TabAll[Nmax, l, m];
   n0 = If[m == 0, 0, 1];
   Do[Export[
     NotebookDirectory[] <> "potential/l_" <> ToString[l] <> "/m_" <> ToString[m] <>
      "/F_l_" <> ToString[l] <> "_m_" <> ToString[m] <> "_n_" <> ToString[n] <> ".hdf5",
     tabAll[[n + 1 - n0]], {"Datasets", "InterpolationTable"}], {n, n0, Nmax}]];
(**)
AutomaticEval[Nmax_, m_, lmin_, lmax_] := Do[Print["l=", l];
   ExportTab[Nmax, l, m], {l, lmin, lmax}];
```

Evaluation

```
m = 2;
lmin = 9;
lmax = 20;
(**)
AutomaticEval[Nb, m, lmin, lmax]
```