In[123]:=
```
(* Mathematica notebook for analytic caculation of the overlap reduction function
  overlap_LM (f,t) for detector pairs constructed from the Hanford, Livingston, Virgo,
 GEO, and TAMA detectors. Follows conventions and definitions in Allen-Ottewill,
 PRD 56, 545 1997 *)
```

In[124]:=
```
(* choose detector pair *)

detectorPair = "HL";
(*detectorPair = "HV";*)
(*detectorPair = "HG";*)
(*detectorPair = "HT";*)
(*detectorPair = "LV";*)
(*detectorPair = "LG";*)
(*detectorPair = "LT";*)
(*detectorPair = "VG";*)
(*detectorPair = "VT";*)
(*detectorPair = "GT";*)
```

In[125]:=
```
(* some globally defined parameters *)

Lmax = 30;
calcPrecision = 64; (* precision for initial numerical calculations *)
Off[General::spell]
```

In[127]:=
```
(* WGS-84 ellipsoidal model of the Earth *)

(* square of semimajor axis (equatorial radius) *)
a2 = 6378137 ^2;
(* square of semiminor axis (polar radius) *)
b2 = (6356752 + 314 / 1000) ^2;
```

In[129]:=
```
(* get location and orientation information for the first detector *)

Switch[StringTake[detectorPair, {1}],

  "H",
  {(* Hanford detector *)
   Print[StringForm["Hanford"]];

   lat = (46 + (27 + (18 + 528 / 1000) / 60) / 60) * Pi / 180;
   lon = -(119 + (24 + (27 + 5657 / 10000) / 60) / 60) * Pi / 180;
   height = 142 + 554 / 1000;

   xarmAz = -(35 + 9994 / 10000) * Pi / 180;
   xarmAlt = -(6 + 195 / 1000) * 10 ^(-4);

   yarmAz = (180 + 54 + 6 / 10000) * Pi / 180;
   yarmAlt = (1 + 25 / 100) * 10 ^(-5);},

  "L",
  {(* Livingston detector *)
   Print[StringForm["Livingston"]];
```

```
  lat = (30 + (33 + (46 + 4196 / 10000) / 60) / 60) * Pi / 180;
  lon = -(90 + (46 + (27 + 2654 / 10000) / 60) / 60) * Pi / 180;
  height = -(6 + 574 / 1000);

  xarmAz = (180 + (72 + 2835 / 10000)) * Pi / 180;
  xarmAlt = -(3 + 121 / 1000) * 10^(-4);

  yarmAz = (180 - (17 + 7165 / 10000)) * Pi / 180;
  yarmAlt = -(6 + 107 / 1000) * 10^(-4);},

 "V",
 {(* Virgo detector *)
  Print[StringForm["Virgo"]];

  lat = (43 + (37 + (53 + 921 / 10000) / 60) / 60) * Pi / 180;
  lon = (10 + (30 + (16 + 1878 / 10000) / 60) / 60) * Pi / 180;
  height = 51 + 884 / 1000;

  xarmAz = (90 - (70 + 5674 / 10000)) * Pi / 180;
  xarmAlt = 0;

  yarmAz = (90 - (160 + 5674 / 10000)) * Pi / 180;
  yarmAlt = 0;},

 "G",
 {(* GEO detector *)
  Print[StringForm["GEO"]];

  lat = (52 + (14 + (42 + 528 / 1000) / 60) / 60) * Pi / 180;
  lon = (9 + (48 + (25 + 894 / 1000) / 60) / 60) * Pi / 180;
  height = 114 + 425 / 1000;

  xarmAz = (90 - (21 + 6117 / 10000)) * Pi / 180;
  xarmAlt = 0;

  yarmAz = (90 - (115 + 9431 / 10000)) * Pi / 180;
  yarmAlt = 0;},

 "T",
 {(* TAMA detector *)
  Print[StringForm["TAMA"]];

  lat = (35 + (40 + (35 + 6 / 10) / 60) / 60) * Pi / 180;
  lon = (139 + (32 + (9 + 8 / 10) / 60) / 60) * Pi / 180;
  height = 90;

  xarmAz = (90 - 180) * Pi / 180;
  xarmAlt = 0;

  yarmAz = (90 - 270) * Pi / 180;
  yarmAlt = 0;}

];
```

Hanford

In[130]:=

```
(* calculate relevant geometric quantities for the first detector *)

R = Sqrt[a2 * Cos[lat] ^ 2 + b2 * Sin[lat] ^ 2];
R1 = (a2 / R + height) * Cos[lat];
R2 = (b2 / R + height) * Sin[lat];

x1 = {R1 * Cos[lon], R1 * Sin[lon] , R2} ;

uloc = {Cos[xarmAlt] * Sin[xarmAz], Cos[xarmAlt] * Cos[xarmAz], Sin[xarmAlt]};
vloc = {Cos[yarmAlt] * Sin[yarmAz], Cos[yarmAlt] * Cos[yarmAz], Sin[yarmAlt]};

Rcylloc = {{0, -Sin[lat], Cos[lat]}, {1, 0, 0}, {0, Cos[lat], Sin[lat]}};
Rcarcyl = {{Cos[lon], -Sin[lon], 0}, {Sin[lon], Cos[lon], 0}, {0, 0, 1}};

u1 = (Rcarcyl . Rcylloc) . uloc;
v1 = (Rcarcyl . Rcylloc ) . vloc;
d1 = (Outer[Times, u1, u1] - Outer[Times, v1, v1]) / 2;

Print[StringForm["x1=``", MatrixForm[N[x1]]]];
Print[StringForm["u1=``", MatrixForm[N[u1]]]];
Print[StringForm["v1=``", MatrixForm[N[v1]]]];
Print[StringForm["d1=``", MatrixForm[N[d1]]]];
```

$$x1=\begin{pmatrix} -2.16141\times 10^6 \\ -3.8347\times 10^6 \\ 4.60035\times 10^6 \end{pmatrix}$$

$$u1=\begin{pmatrix} -0.223893 \\ 0.799831 \\ 0.556905 \end{pmatrix}$$

$$v1=\begin{pmatrix} -0.913978 \\ 0.026094 \\ -0.404923 \end{pmatrix}$$

$$d1=\begin{pmatrix} -0.392614 & -0.0776134 & -0.247389 \\ -0.0776134 & 0.319524 & 0.227998 \\ -0.247389 & 0.227998 & 0.07309 \end{pmatrix}$$

In[145]:=

```
(* get location and orientation information for the second detector *)

Switch[StringTake[detectorPair, {2}],

  "H",
  {(* Hanford detector *)
   Print[StringForm["Hanford"]];

   lat = (46 + (27 + (18 + 528 / 1000) / 60) / 60) * Pi / 180;
   lon = -(119 + (24 + (27 + 5657 / 10000) / 60) / 60) * Pi / 180;
   height = 142 + 554 / 1000;

   xarmAz = -(35 + 9994 / 10000) * Pi / 180;
   xarmAlt = -(6 + 195 / 1000) * 10 ^ (-4);

   yarmAz = (180 + 54 + 6 / 10000) * Pi / 180;
   yarmAlt = (1 + 25 / 100) * 10 ^ (-5);},
```

```
"L",
{(* Livingston detector *)
 Print[StringForm["Livingston"]];

 lat = (30 + (33 + (46 + 4196 / 10000) / 60) / 60) * Pi / 180;
 lon = -(90 + (46 + (27 + 2654 / 10000) / 60) / 60) * Pi / 180;
 height = -(6 + 574 / 1000);

 xarmAz = (180 + (72 + 2835 / 10000)) * Pi / 180;
 xarmAlt = -(3 + 121 / 1000) * 10^(-4);

 yarmAz = (180 - (17 + 7165 / 10000)) * Pi / 180;
 yarmAlt = -(6 + 107 / 1000) * 10^(-4);},

"V",
{(* Virgo detector *)
 Print[StringForm["Virgo"]];

 lat = (43 + (37 + (53 + 921 / 10000) / 60) / 60) * Pi / 180;
 lon = (10 + (30 + (16 + 1878 / 10000) / 60) / 60) * Pi / 180;
 height = 51 + 884 / 1000;

 xarmAz = (90 - (70 + 5674 / 10000)) * Pi / 180;
 xarmAlt = 0;

 yarmAz = (90 - (160 + 5674 / 10000)) * Pi / 180;
 yarmAlt = 0;},

"G",
{(* GEO detector *)
 Print[StringForm["GEO"]];

 lat = (52 + (14 + (42 + 528 / 1000) / 60) / 60) * Pi / 180;
 lon = (9 + (48 + (25 + 894 / 1000) / 60) / 60) * Pi / 180;
 height = 114 + 425 / 1000;

 xarmAz = (90 - (21 + 6117 / 10000)) * Pi / 180;
 xarmAlt = 0;

 yarmAz = (90 - (115 + 9431 / 10000)) * Pi / 180;
 yarmAlt = 0;},

"T",
{(* TAMA detector *)
 Print[StringForm["TAMA"]];

 lat = (35 + (40 + (35 + 6 / 10) / 60) / 60) * Pi / 180;
 lon = (139 + (32 + (9 + 8 / 10) / 60) / 60) * Pi / 180;
 height = 90;

 xarmAz = (90 - 180) * Pi / 180;
 xarmAlt = 0;

 yarmAz = (90 - 270) * Pi / 180;
 yarmAlt = 0;}

];
```

Livingston

In[146]:=

```
(* calculate relevant geometric quantities for the second detector *)

R = Sqrt[a2 * Cos[lat]^2 + b2 * Sin[lat]^2];
R1 = (a2 / R + height) * Cos[lat];
R2 = (b2 / R + height) * Sin[lat];

x2 = {R1 * Cos[lon], R1 * Sin[lon] , R2} ;

uloc = {Cos[xarmAlt] * Sin[xarmAz], Cos[xarmAlt] * Cos[xarmAz], Sin[xarmAlt]};
vloc = {Cos[yarmAlt] * Sin[yarmAz], Cos[yarmAlt] * Cos[yarmAz], Sin[yarmAlt]};

Rcylloc = {{0, -Sin[lat], Cos[lat]}, {1, 0, 0}, {0, Cos[lat], Sin[lat]}};
Rcarcyl = {{Cos[lon], -Sin[lon], 0}, {Sin[lon], Cos[lon], 0}, {0, 0, 1}};

u2 = Rcarcyl . Rcylloc . uloc;
v2 = Rcarcyl . Rcylloc . vloc;
d2 = (Outer[Times, u2, u2] - Outer[Times, v2, v2]) / 2;

Print[StringForm["x2=``", MatrixForm[N[x2]]]];
Print[StringForm["u2=``", MatrixForm[N[u2]]]];
Print[StringForm["v2=``", MatrixForm[N[v2]]]];
Print[StringForm["d2=``", MatrixForm[N[d2]]]];
```

$$x2=\begin{pmatrix} -74276. \\ -5.49628\times 10^{6} \\ 3.22426\times 10^{6} \end{pmatrix}$$

$$u2=\begin{pmatrix} -0.954574 \\ -0.141581 \\ -0.262189 \end{pmatrix}$$

$$v2=\begin{pmatrix} 0.297742 \\ -0.48791 \\ -0.820545 \end{pmatrix}$$

$$d2=\begin{pmatrix} 0.411281 & 0.14021 & 0.247295 \\ 0.14021 & -0.109006 & -0.181616 \\ 0.247295 & -0.181616 & -0.302275 \end{pmatrix}$$

In[161]:=

```
(* Rotate to Earth-based frame where separation vector has no y-component *)

DeltaX = x1 - x2;
phi = ArcTan[DeltaX[[2]] / DeltaX[[1]]];

Print[StringForm["phi=`` degrees", N[phi * 180 / Pi]]];
Rot1 = { {Cos[phi], Sin[phi], 0}, {-Sin[phi], Cos[phi], 0}, {0, 0, 1}};

x1bar = Rot1.x1;
u1bar = Rot1.u1;
v1bar = Rot1.v1;
d1bar = (Outer[Times, u1bar, u1bar] - Outer[Times, v1bar, v1bar]) / 2;

x2bar = Rot1.x2;
u2bar = Rot1.u2;
v2bar = Rot1.v2;
d2bar = (Outer[Times, u2bar, u2bar] - Outer[Times, v2bar, v2bar]) / 2;

DeltaXbar = x1bar - x2bar;

Print[StringForm["x1bar =``", MatrixForm[N[x1bar]]]];
Print[StringForm["u1bar =``", MatrixForm[N[u1bar]]]];
Print[StringForm["v1bar =``", MatrixForm[N[v1bar]]]];
Print[StringForm["d1bar =``", MatrixForm[N[d1bar]]]];
Print[StringForm["x2bar =``", MatrixForm[N[x2bar]]]];
Print[StringForm["u2bar =``", MatrixForm[N[u2bar]]]];
Print[StringForm["v2bar =``", MatrixForm[N[v2bar]]]];
Print[StringForm["d2bar =``", MatrixForm[N[d2bar]]]];
Print[StringForm["DeltaXbar =``", MatrixForm[N[DeltaXbar]]]];

(* set y-component of DeltaXbar identically to zero;
 was non-zero due to machine precision *)
DeltaXbar[[2]] = 0;
Print[StringForm["DeltaXbar =``", MatrixForm[N[DeltaXbar]]]];

(* compare with Allen-Ottewill, between Eqs. 6.2, 6.3 *)
```

```
phi=-38.5236 degrees
```

$$\text{x1bar} = \begin{pmatrix} 697402. \\ -4.34629 \times 10^6 \\ 4.60035 \times 10^6 \end{pmatrix}$$

$$\text{u1bar} = \begin{pmatrix} -0.673327 \\ 0.4863 \\ 0.556905 \end{pmatrix}$$

$$\text{v1bar} = \begin{pmatrix} -0.731305 \\ -0.548844 \\ -0.404923 \end{pmatrix}$$

$$\text{d1bar} = \begin{pmatrix} -0.0407189 & -0.364406 & -0.335551 \\ -0.364406 & -0.0323711 & 0.0242915 \\ -0.335551 & 0.0242915 & 0.07309 \end{pmatrix}$$

$$\text{x2bar} = \begin{pmatrix} 3.36518 \times 10^6 \\ -4.34629 \times 10^6 \\ 3.22426 \times 10^6 \end{pmatrix}$$

$$\text{u2bar} = \begin{pmatrix} -0.658631 \\ -0.70531 \\ -0.262189 \end{pmatrix}$$

$$\text{v2bar} = \begin{pmatrix} 0.536827 \\ -0.196273 \\ -0.820545 \end{pmatrix}$$

$$\text{d2bar} = \begin{pmatrix} 0.0728058 & 0.284952 & 0.306588 \\ 0.284952 & 0.229469 & 0.0119368 \\ 0.306588 & 0.0119368 & -0.302275 \end{pmatrix}$$

$$\text{DeltaXbar} = \begin{pmatrix} -2.66778 \times 10^6 \\ -4.65661 \times 10^{-10} \\ 1.37609 \times 10^6 \end{pmatrix}$$

$$\text{DeltaXbar} = \begin{pmatrix} -2.66778 \times 10^6 \\ 0. \\ 1.37609 \times 10^6 \end{pmatrix}$$

In[184]:=

```
(* Rotate to Earth-
  based calculation frame where separation vector points in the z-direction *)

beta = ArcTan[DeltaXbar[[1]] / DeltaXbar[[3]]];

Print[StringForm["beta=`` degrees", N[beta * 180 / Pi]]];
Rot2 = {{Cos[beta], 0, -Sin[beta]}, {0, 1, 0}, {Sin[beta], 0, Cos[beta]}};

x1prime = Rot2.x1bar;
u1prime = Rot2.u1bar;
v1prime = Rot2.v1bar;
d1prime = (Outer[Times, u1prime, u1prime] - Outer[Times, v1prime, v1prime]) / 2;

x2prime = Rot2.x2bar;
u2prime = Rot2.u2bar;
v2prime = Rot2.v2bar;
d2prime = (Outer[Times, u2prime, u2prime] - Outer[Times, v2prime, v2prime]) / 2;

DeltaXprime = x1prime - x2prime;

Print[StringForm["x1prime =``", MatrixForm[N[x1prime]]]];
Print[StringForm["u1prime =``", MatrixForm[N[u1prime]]]];
Print[StringForm["v1prime =``", MatrixForm[N[v1prime]]]];
Print[StringForm["d1prime =``", MatrixForm[N[d1prime]]]];
Print[StringForm["x2prime =``", MatrixForm[N[x2prime]]]];
Print[StringForm["u2prime =``", MatrixForm[N[u2prime]]]];
Print[StringForm["v2prime =``", MatrixForm[N[v2prime]]]];
Print[StringForm["d2prime =``", MatrixForm[N[d2prime]]]];
Print[StringForm["DeltaXprime =``", MatrixForm[N[DeltaXprime]]]];

(* set x,y-components of DeltaXprime identically to zero;
 were non-zero due to machine precision *)
DeltaXprime[[1]] = 0;
DeltaXprime[[2]] = 0;
Print[StringForm["DeltaXprime =``", MatrixForm[N[DeltaXprime]]]];

(* compare with Allen-Ottewill Eq. 6.6 *)
```

```
beta=-62.7144 degrees
```

$$\text{x1prime} = \begin{pmatrix} 4.40819 \times 10^6 \\ -4.34629 \times 10^6 \\ 1.48912 \times 10^6 \end{pmatrix}$$

$$\text{u1prime} = \begin{pmatrix} 0.186269 \\ 0.4863 \\ 0.853707 \end{pmatrix}$$

$$\text{v1prime} = \begin{pmatrix} -0.695118 \\ -0.548844 \\ 0.464307 \end{pmatrix}$$

$$\text{d1prime} = \begin{pmatrix} -0.224247 & -0.145465 & 0.240883 \\ -0.145465 & -0.0323711 & 0.334995 \\ 0.240883 & 0.334995 & 0.256618 \end{pmatrix}$$

$$\text{x2prime} = \begin{pmatrix} 4.40819 \times 10^{6} \\ -4.34629 \times 10^{6} \\ -1.51266 \times 10^{6} \end{pmatrix}$$

$$\text{u2prime} = \begin{pmatrix} -0.53495 \\ -0.70531 \\ 0.465152 \end{pmatrix}$$

$$\text{v2prime} = \begin{pmatrix} -0.483149 \\ -0.196273 \\ -0.853255 \end{pmatrix}$$

$$\text{d2prime} = \begin{pmatrix} 0.0263693 & 0.141238 & -0.330541 \\ 0.141238 & 0.229469 & -0.247774 \\ -0.330541 & -0.247774 & -0.255839 \end{pmatrix}$$

$$\text{DeltaXprime} = \begin{pmatrix} -9.31323 \times 10^{-10} \\ -4.65661 \times 10^{-10} \\ 3.00178 \times 10^{6} \end{pmatrix}$$

$$\text{DeltaXprime} = \begin{pmatrix} 0. \\ 0. \\ 3.00178 \times 10^{6} \end{pmatrix}$$

In[207]:=

```
(* calculate s_k (u), non-zero for Abs[k]≤4 *)
Clear[u];

omegaPrime =
  {Sin[thetaPrime] Cos[phiPrime], Sin[thetaPrime] Sin[phiPrime], Cos[thetaPrime]};
omegaPrimeAB = Outer[Times, omegaPrime, omegaPrime];
omegaPrimeABCD = Outer[Times, omegaPrime, omegaPrime, omegaPrime, omegaPrime];

For[K = 0, K ≤ Lmax, K++,

 If[K > 4,
  s[K, u_] = 0,
  { (* else *)
   Print[StringForm["Calculating `` out of ``", K, 4]];
   int = Integrate[Exp[Sqrt[-1] * K * phiPrime], {phiPrime, 0, 2 * Pi}];
   intAB = Integrate[Exp[Sqrt[-1] * K * phiPrime] * omegaPrimeAB,
      {phiPrime, 0, 2 * Pi}] /. {Cos[thetaPrime] → u, Sin[thetaPrime] → Sqrt[1 - u^2]};
   intABCD = Integrate[Exp[Sqrt[-1] * K * phiPrime] * omegaPrimeABCD,
      {phiPrime, 0, 2 * Pi}] /. {Cos[thetaPrime] → u, Sin[thetaPrime] → Sqrt[1 - u^2]};

   (*Print[MatrixForm[intAB]]; Print[MatrixForm[intABCD]];*)

   M1 = N[int * d1prime.d2prime, calcPrecision] ;
   M2 = N[d1prime .intAB . d2prime, calcPrecision];
   M3 = N[d1prime . intABCD . d2prime, calcPrecision];

   s[K, u_] = FullSimplify[
     2 * Tr[M1] - 4 * Tr[M2] + Tr[M3[[1, 1]]] + Tr[M3[[2, 2]]] + Tr[M3[[3, 3]]]];
  }
 ]

]

(* calculate s_k (u) for negative k *)
For[K = -1, K >= -Lmax, K--,
 s[K, u_] = Conjugate[s[-K, u]];
]
```

```
Calculating 0 out of 4

Calculating 1 out of 4

Calculating 2 out of 4

Calculating 3 out of 4

Calculating 4 out of 4
```

In[213]:=

```
(* display non-zero s_k(u), compare with Allen-Ottewill, between Eqs. 8.3, 8.4 *)

For[K = -4, K ≤ 4, K++,
 Print[StringForm["s``[u]=``", K, N[s[K, u]]]];
]
```

$$s_{-4}[u] = (0.0475757 - 0.00191932\,i)\,(1. - 1.\,\text{Conjugate}[u]^2)^2$$

$$s_{-3}[u] = (-0.239096 - 0.225742\,i)\,\text{Conjugate}\left[u\,(1. - 1.\,u^2)^{3/2}\right]$$

$$s_{-2}[u] = (-0.0283834 - 1.41691\,i) + \text{Conjugate}[(0.0424886 - 2.14149\,i)\,u^2 - (0.0141052 - 0.724574\,i)\,u^4]$$

$$s_{-1}[u] = ((0.526317 - 0.536609\,i) - (1.66491 - 1.69714\,i)\,\text{Conjugate}[u]^2)\,\text{Conjugate}\left[u\sqrt{1. + 0.\,u - 1.\,u^2}\right]$$

$$s_0[u] = -3.00569 + 1.65266\,u^2 + 1.08152\,u^4$$

$$s_1[u] = u\,((0.526317 + 0.536609\,i) - (1.66491 + 1.69714\,i)\,u^2)\,\sqrt{1. + 0.\,u - 1.\,u^2}$$

$$s_2[u] = (-0.0283834 + 1.41691\,i) + (0.0424886 - 2.14149\,i)\,u^2 - (0.0141052 - 0.724574\,i)\,u^4$$

$$s_3[u] = (-0.239096 + 0.225742\,i)\,u\,(1. - 1.\,u^2)^{3/2}$$

$$s_4[u] = (0.0475757 + 0.00191932\,i)\,(1. - 1.\,u^2)^2$$

In[214]:=

```
(* extract coefficients a[k,N] of s_k(u) defined by
   s_k(u) = (1-u^2)^(|k|/2) sum_{N=0}^{4-|k|} a[k,N] u^N. *)

For[K = 0, K ≤ Lmax, K++,
 For[NN = 0, NN ≤ 4 - Abs[K], NN++,
  a[K, NN] = D[s[K, u] / (1 - u^2)^(Abs[K] / 2), {u, NN}] / (Factorial[NN]) /. u → 0;
  (*Print[StringForm["a````=``",K,NN, N[a[K,NN]]]];*)
 ]
]

For[K = -1, K >= -Lmax, K--,
 For[NN = 0, NN ≤ 4 - Abs[K], NN++,
  a[K, NN] = Conjugate[a[-K, NN]];
  (*Print[StringForm["a````=``",K,NN, N[a[K,NN]]]];*)
 ]
]
```

In[216]:=

```
(* extract coefficients b[K,N,r,M] of the polynomials that appear
  in the expansion of the derivative of Exp[ixu] (1-u^2)^k u^N *)

Clear[u, x];
For[K = 0, K ≤ Lmax, K++,

 For[NN = 0, NN <= 4 - Abs[K], NN++,

  polytemp[K, NN, x_] =
   FullSimplify[Exp[-Sqrt[-1] * x * u] * D[Exp[Sqrt[-1] * x * u] (1 - u^2)^K u^NN, {u, K}]];
   (*Print[StringForm["poly````[u,x]=``",K,NN, polytemp[K,NN,x]]];*)

  For[r = 0, r ≤ K, r++,

   poly[K, NN, r] = D[polytemp[K, NN, x], {x, r}] / (Sqrt[-1]^r Factorial[r]) /. x → 0;
   (*Print[StringForm["poly``````=``",K,NN,r,poly[K,NN,r]]];*)

   poly[K, NN, r] = Expand[poly[K, NN, r]];
   (*Print[StringForm["poly``````=``",K,NN,r,poly[K,NN,r]]];*)

   For[MM = 0, MM ≤ K + NN + r, MM++,
    b[K, NN, r, MM] = D[poly[K, NN, r], {u, MM}] / (Factorial[MM]) /. u → 0;
    (*Print[StringForm["b````````=``",K,NN,r,MM,b[K,NN,r,MM]]];*)
   ]
  ]
 ]
]

(* see Allen-Ottewill, Eqs. A5, A6 with Exp[iux] factored out *)
```

In[218]:=

```
(* define substitution for derivatives of spherical bessel functions *)

Clear[x];
For[L = -(Lmax + 4), L ≤ 2 * Lmax + 4, L++,
 djsub[L] = D[j[L, x_], x_] → (L / (2 * L + 1)) * j[L - 1, x] - ((L + 1) / (2 * L + 1)) * j[L + 1, x];
]

dsphericalBesselJsubstitution = Table[djsub[i], {i, -(Lmax + 4), 2 * Lmax + 4}];
```

In[221]:=

```
(* calculate JLM integral, Allen-Ottewill Eq. A7 *)

Clear[x];
For[L = 0, L ≤ Lmax, L++,

 Print[StringForm["Calculating `` out of ``", L, Lmax]];

 MM = 0;
 JJ[MM, L, x_] = (2 Sqrt[-1]^L) * j[L, x];
 (*Print[StringForm["J^``_``=``",MM, L, JJ[MM,L,x]]];*)

 For[MM = 1, MM ≤ Lmax + 4, MM++,
  JJ[MM, L, x_] =
   Simplify[(-Sqrt[-1]) D[JJ[MM - 1, L, x], x] //. dsphericalBesselJsubstitution];
  (*Print[StringForm["J^``_``[x]=``",MM,L, JJ[MM,L,x]]];*)
 ]
]

(* display a few JLM's *)

Print[StringForm["J^0_0[x]=``", JJ[0, 0, x]]];
Print[StringForm["J^1_0[x]=``", JJ[1, 0, x]]];
Print[StringForm["J^1_1[x]=``", JJ[1, 1, x]]];
Print[StringForm["J^2_1[x]=``", JJ[2, 1, x]]];

(* See Allen-Ottewill, Eqs. A7-A11. *)
(* The expressions after Eq. A11 are missing a factor of 2 i^L (-i)^M *)
```

```
Calculating 0 out of 30

Calculating 1 out of 30

Calculating 2 out of 30

Calculating 3 out of 30

Calculating 4 out of 30

Calculating 5 out of 30

Calculating 6 out of 30

Calculating 7 out of 30

Calculating 8 out of 30

Calculating 9 out of 30

Calculating 10 out of 30

Calculating 11 out of 30

Calculating 12 out of 30

Calculating 13 out of 30

Calculating 14 out of 30
```

```
Calculating 15 out of 30

Calculating 16 out of 30

Calculating 17 out of 30

Calculating 18 out of 30

Calculating 19 out of 30

Calculating 20 out of 30

Calculating 21 out of 30

Calculating 22 out of 30

Calculating 23 out of 30

Calculating 24 out of 30

Calculating 25 out of 30

Calculating 26 out of 30

Calculating 27 out of 30

Calculating 28 out of 30

Calculating 29 out of 30

Calculating 30 out of 30

J^0_0[x]=2 j[0, x]

J^1_0[x]=2 i j[1, x]

J^1_1[x]=2/3 (j[0, x] - 2 j[2, x])

J^2_1[x]=2/5 i (3 j[1, x] - 2 j[3, x])
```

In[227]:=

```
(* calculate ILKN integral,
 see Allen-Ottewill Eq. A5 including normalisation factor Eq. 7.9 *)
Clear[x];

For[L = 0, L ≤ Lmax, L++,
 Print[StringForm["Calculating `` out of ``", L, Lmax]];

 For[K = 0, K ≤ L, K++,

  Normfactor[L, K] = Sqrt[((2 L + 1) / (4 Pi)) * (Factorial[L - K] / Factorial[L + K])];

  For[NN = 0, NN <= 4 - Abs[K], NN++,

   II[L, K, NN, x_] = 0;
   For[r = 0, r ≤ K, r++,

    sumM[L, K, NN, r, x_] = 0;
    For[MM = 0, MM ≤ K + NN + r, MM++,
     sumM[L, K, NN, r, x_] = sumM[L, K, NN, r, x] + b[K, NN, r, MM] * JJ[MM, L, x];
    ];

    II[L, K, NN, x_] =
     II[L, K, NN, x] + Normfactor[L, K] (Sqrt[-1] x)^r sumM[L, K, NN, r, x];
   ];

   (* assign values for negative K *)
   If[K > 0,
    II[L, -K, NN, x_] = (-1)^K II[L, K, NN, x]
   ];

   (*Print[StringForm["N=``, I^``_``[x]=``",NN,K,L,Simplify[II[L,K,NN,x]]]];*)

  ]
 ]
]
```

```
Calculating 0 out of 30

Calculating 1 out of 30

Calculating 2 out of 30

Calculating 3 out of 30

Calculating 4 out of 30

Calculating 5 out of 30

Calculating 6 out of 30

Calculating 7 out of 30

Calculating 8 out of 30

Calculating 9 out of 30

Calculating 10 out of 30
```

```
Calculating 11 out of 30

Calculating 12 out of 30

Calculating 13 out of 30

Calculating 14 out of 30

Calculating 15 out of 30

Calculating 16 out of 30

Calculating 17 out of 30

Calculating 18 out of 30

Calculating 19 out of 30

Calculating 20 out of 30

Calculating 21 out of 30

Calculating 22 out of 30

Calculating 23 out of 30

Calculating 24 out of 30

Calculating 25 out of 30

Calculating 26 out of 30

Calculating 27 out of 30

Calculating 28 out of 30

Calculating 29 out of 30

Calculating 30 out of 30
```

In[229]:=

```
(* calculate rotation matrix components dLMK, see Allen-Ottewill Eq. 7.6 *)

beta = N[beta, calcPrecision];

For[L = 0, L ≤ Lmax, L++,

 Print[StringForm["Calculating `` out of ``", L, Lmax]];

 For[M = 0, M ≤ L, M++,
  For[K = -L, K ≤ L, K++,

   q = Min[L + K, L - K, L + M, L - M];

   Which[
    q == L + K,
    d[L, M, K] = (-1)^(M - K)
      Sqrt[(Factorial[L - K] Factorial[L + K]) / (Factorial[L + M] Factorial[L - M])]
      (Sin[beta / 2])^(M - K) (Cos[beta / 2])^(-M - K)  JacobiP[L + K, M - K, -M - K, Cos[beta]]

    q == L - K,
    d[L, M, K] = Sqrt[(Factorial[L + K] Factorial[L - K]) / (Factorial[L + M] Factorial[L - M])
      (Sin[beta / 2])^(K - M) (Cos[beta / 2])^(M + K)  JacobiP[L - K, K - M, M + K, Cos[beta]],

    q == L + M,
    d[L, M, K] = Sqrt[(Factorial[L - M] Factorial[L + M]) / (Factorial[L + K] Factorial[L - K])
      (Sin[beta / 2])^(K - M) (Cos[beta / 2])^(-K - M)  JacobiP[L + M, K - M, -K - M, Cos[beta]]

    q == L - M,
    d[L, M, K] = (-1)^(M - K)
      Sqrt[(Factorial[L + M] Factorial[L - M]) / (Factorial[L + K] Factorial[L - K])]
      (Sin[beta / 2])^(M - K) (Cos[beta / 2])^(K + M)  JacobiP[L - M, M - K, K + M, Cos[beta]]

   ]

  ]
 ]
]

(* display a few dLMK's *)

For[L = 0, L ≤ 1, L++,
 For[M = 0, M ≤ L, M++,
  For[K = -L, K ≤ L, K++,
   Print[StringForm["d``````=``", L, M, K, N[d[L, M, K]]]];
  ]
 ]
]
```

```
Calculating 0 out of 30

Calculating 1 out of 30

Calculating 2 out of 30
```

```
Calculating 3 out of 30

Calculating 4 out of 30

Calculating 5 out of 30

Calculating 6 out of 30

Calculating 7 out of 30

Calculating 8 out of 30

Calculating 9 out of 30

Calculating 10 out of 30

Calculating 11 out of 30

Calculating 12 out of 30

Calculating 13 out of 30

Calculating 14 out of 30

Calculating 15 out of 30

Calculating 16 out of 30

Calculating 17 out of 30

Calculating 18 out of 30

Calculating 19 out of 30

Calculating 20 out of 30

Calculating 21 out of 30

Calculating 22 out of 30

Calculating 23 out of 30

Calculating 24 out of 30

Calculating 25 out of 30

Calculating 26 out of 30

Calculating 27 out of 30

Calculating 28 out of 30

Calculating 29 out of 30

Calculating 30 out of 30

d000=1.`

d10-1=0.6284286951552364`

d100=0.4584263847238442`

d101=-0.628429
```

```
d11-1=0.2707868076380779`

d110=0.6284286951552364`

d111=0.7292131923619221`
```

In[232]:=

```
(* calculate INITIAL decomposition of
  overlapLM[x] in terms of spherical bessel functions *)
Clear[x];

For[L = 0, L ≤ Lmax, L++,
 Print[StringForm["Calculating `` out of ``", L, Lmax]];
 For[M = 0, M ≤ L, M++,

  overlap[L, M, x_] = 0;
  For[K = -L, K ≤ L, K++,

   sumN[L, K, x_] = 0;
   For[NN = 0, NN <= 4 - Abs[K], NN++,
    sumN[L, K, x_] = sumN[L, K, x] + a[K, NN] * II[L, K, NN, x];
   ];

   overlap[L, M, x_] = overlap[L, M, x] + (5 / (8 * Pi)) * d[L, M, K] * sumN[L, K, x];

  ];

  overlap[L, M, x_] = Simplify[overlap[L, M, x]];
  (*Print[StringForm["overlap````[x]=``",L,M, N[overlap[L,M,x]]]];*)
 ]
]
```

```
Calculating 0 out of 30

Calculating 1 out of 30

Calculating 2 out of 30

Calculating 3 out of 30

Calculating 4 out of 30

Calculating 5 out of 30

Calculating 6 out of 30

Calculating 7 out of 30

Calculating 8 out of 30

Calculating 9 out of 30

Calculating 10 out of 30

Calculating 11 out of 30

Calculating 12 out of 30
```

```
Calculating 13 out of 30

Calculating 14 out of 30

Calculating 15 out of 30

Calculating 16 out of 30

Calculating 17 out of 30

Calculating 18 out of 30

Calculating 19 out of 30

Calculating 20 out of 30

Calculating 21 out of 30

Calculating 22 out of 30

Calculating 23 out of 30

Calculating 24 out of 30

Calculating 25 out of 30

Calculating 26 out of 30

Calculating 27 out of 30

Calculating 28 out of 30

Calculating 29 out of 30

Calculating 30 out of 30
```

In[234]:=

```
(* define spherical bessel function substitutions *)
Clear[x];

(* note that Lmax+4 = max value of MM *)
For[L = 0, L ≤ 2 * Lmax + 4, L++,
 (* construct spherical bessel functions of the 3 rd kind h1[n,x] for n≥0 *)
 h1[L] = 0;
 For[K = 0, K ≤ L, K++,
  h1[L] = h1[L] + Factorial[L + K] / (Factorial[K] Gamma[L - K + 1]) * (-2 * Sqrt[-1] * x) ^ (-K);
 ];
 h1[L] = Sqrt[-1] ^ (-L - 1) * x ^ (-1) * Exp[Sqrt[-1] x] * h1[L];

 (* spherical bessel function j[n,x] is real part of h1[n,x] *)
 jsub[L] = j[L, x_] → Re[h1[L]];

 (*Print[StringForm["j``[x]=``",L, ComplexExpand[Re[h1[L]]]]];*)
]

sphericalBesselJsubstitution = Table[jsub[i], {i, 0, 2 * Lmax + 4}];
```

In[237]:=

```
(* extract coefficients of polynomials in 1/x
  multiplying Sin[x] and Cos[x] for spherical bessel functions *)
Clear[x];

For[L = 0, L ≤ 2 * Lmax + 4, L++,
 If[Mod[L, 10] == 0, Print[StringForm["Calculating `` out of ``", L, 2 * Lmax + 4]]];

 poly[x_] = ExpandAll[ComplexExpand[Re[h1[L]]]] //. {Cos[x] → 1, Sin[x] → 1};

 Clear[r];
 rMax = L + 1;
 For[r = rMax, r ≥ 1, r--,
  pp[x_] = Expand[x^rMax poly[x]];
  polyJCoeff[L, r] = (D[pp[x], {x, rMax - r}] / Factorial[rMax - r]) /. x → 0;
 ]
]

(* display *)
(*
 For[L=0, L≤Lmax, L++,
  Print[StringForm["j``[x]=``",L,ExpandAll[ComplexExpand[Re[h1[L]]]]]];
  rMax=L+1;
  For[r=rMax, r≥1, r--,
   Print[StringForm["j````=``",L,r,polyJCoeff[L,r]]];
  ]
 ]
*)
```

```
Calculating 0 out of 64

Calculating 10 out of 64

Calculating 20 out of 64

Calculating 30 out of 64

Calculating 40 out of 64

Calculating 50 out of 64

Calculating 60 out of 64
```

In[239]:=

```
(* extract coefficients of polynomials in 1/x
  multiplying Sin[x] and Cos[x] for overlap reduction function *)
Clear[x];

For[L = 0, L ≤ Lmax, L++,
 Print[StringForm["Calculating `` out of ``", L, Lmax]];
 For[M = 0, M ≤ L, M++,

  ff[x_] = overlap[L, M, x] //. sphericalBesselJsubstitution;
  poly[x_] = ExpandAll[ComplexExpand[ff[x]]] //. {Cos[x] → 1, Sin[x] -> 1};

  Clear[r];
  rMax = L + 5;
  For[r = rMax, r ≥ 1, r--,
   pp[x_] = Expand[x^rMax poly[x]];
   polyOverlapCoeff[L, M, r] = (D[pp[x], {x, rMax - r}] / Factorial[rMax - r]) /. x → 0;
  ]
 ]
]

(* display *)
(*
 For[L=0, L≤1, L++,
  For[M=0, M≤L, M++,
   Print[StringForm["overlap````[x]=``",L,M,
     ExpandAll[ComplexExpand[overlap[L,M,x]//.sphericalBesselJsubstitution]]]];
   rMax=L+5;
   For[r=rMax, r≥1, r--,
    Print[StringForm["overlap``````=``",L,M,r,N[polyOverlapCoeff[L,M,r]]]];
   ]
  ]
 ]
*)
```

```
Calculating 0 out of 30

Calculating 1 out of 30

Calculating 2 out of 30

Calculating 3 out of 30

Calculating 4 out of 30

Calculating 5 out of 30

Calculating 6 out of 30

Calculating 7 out of 30

Calculating 8 out of 30

Calculating 9 out of 30

Calculating 10 out of 30

Calculating 11 out of 30
```

```
Calculating 12 out of 30

Calculating 13 out of 30

Calculating 14 out of 30

Calculating 15 out of 30

Calculating 16 out of 30

Calculating 17 out of 30

Calculating 18 out of 30

Calculating 19 out of 30

Calculating 20 out of 30

Calculating 21 out of 30

Calculating 22 out of 30

Calculating 23 out of 30

Calculating 24 out of 30

Calculating 25 out of 30

Calculating 26 out of 30

Calculating 27 out of 30

Calculating 28 out of 30

Calculating 29 out of 30
```

```
Calculating 30 out of 30
```

In[241]:=

```
(* calculate coefficients of an alternative spherical bessel function decomposition
  of the overlap reduction function, as in Allen-Ottewill after Eq. 8.4 *)

(* NOTE: only need to solve the matrix
   equation for the polynomial coefficients multiplying Sin[x] *)

(* OCoeff = JMatrix * JCoeff *)

For[L = 0, L ≤ Lmax, L++,
 For[M = 0, M ≤ L, M++,

  matrixDimension = 2 + Floor[(L + 2) / 2];
  highestN = 2 + Floor[(L + 1) / 2];

  (* construct vector of original overlap reduction function polynomial coeffs *)
  OCoeff = Table[polyOverlapCoeff[L, M, r], {r, L + 5, 1, -2}];
  (*Print[StringForm["OCoeff````=``",L,M,MatrixForm[OCoeff]]];*)

  (* construct JMatrix *)
  JMatrix = Table[0, {i, 1, matrixDimension}, {j, 1, matrixDimension}];

  (* work backwards from rightmost column to leftmost column *)
  For[col = matrixDimension, col ≥ 1, col--,

   NN = highestN - (matrixDimension - col);

   For[row = 1 + (matrixDimension - col), row ≤ matrixDimension, row++,
    r = NN + 1 - 2 * (row - (1 + (matrixDimension - col)));
    (*Print[StringForm["col=``, row=``, N=``, r=``",col,row,NN,r]];*)

    If[r ≥ 1,
     JMatrix[[row, col]] = polyJCoeff[NN, r]
    ]

   ]

  ];

  (*Print[StringForm["JMatrix````=``",L,M,MatrixForm[JMatrix]]];*)

  (* invert matrix equation OCoeff = JMatrix * JCoeff to find JCoeff *)
  JInvMatrix = Inverse[JMatrix];

  JCoeff[L, M] = JInvMatrix . OCoeff;
  (* reduce precision to 16 digits *)
  JCoeff[L, M] = N[JCoeff[L, M], 16];
 ]
]
```

In[242]:=

```
(* display a couple of the new overlap reduction function coefficients *)

For[L = 0, L ≤ 1, L++,
 For[M = 0, M ≤ L, M++,
  Print[StringForm["JCoeff````=``", L, M, MatrixForm[JCoeff[L, M]]]];
 ]
]
```

$$\text{JCoeff00}= \begin{pmatrix} -0.03047375076852364 \\ -0.8565652425169227 \\ 0.9711388346235925 \end{pmatrix}$$

$$\text{JCoeff10}= \begin{pmatrix} 0.\times 10^{-18} - 0.02419670023432767\,i \\ 0.\times 10^{-17} - 1.0735727969032796\,i \\ 0.\times 10^{-16} + 3.072365652877589\,i \end{pmatrix}$$

$$\text{JCoeff11}= \begin{pmatrix} 0.\times 10^{-18} - 0.03316977657051953\,i \\ -0.3190703652327734 - 0.7888408276532609\,i \\ 1.866412498835104 + 0.217692417677806\,i \end{pmatrix}$$

In[243]:=

```
(* write overlap reduction function coefficients to files *)

SetDirectory["~/src/matapps/src/searches/stochastic/CrossCorr/"];

For[L = 0, L ≤ Lmax, L++,
 For[M = 0, M ≤ L, M++,

  filename = detectorPair <> "jcoeffs" <> ToString[L] <> ToString[M] <> "real.dat";
  Export[filename, N[Re[JCoeff[L, M]], 16], "List"];

  filename = detectorPair <> "jcoeffs" <> ToString[L] <> ToString[M] <> "imag.dat";
  Export[filename, N[Im[JCoeff[L, M]], 16], "List"];

 ]
]
```