# Basic functions

In[21]:=
```
GaussianIntegralMatrix[M_] :=
```
$$\lim_{\{m13,m31,m22\}\to\{M[[1,3]],M[[3,1]],M[[2,2]]\}} \frac{2\, e^{\frac{M[[1,3]]\, M[[2,1]]^2 - M[[1,2]]\, M[[2,1]]\, M[[2,2]] + M[[1,1]]\, M[[2,2]]^2 + M[[1,2]]^2\, M[[3,1]] - 4\, M[[1,1]]\, M[[1,3]]\, M[[3,1]]}{m22^2 - 4\, m13\, m31}}\, \pi}{\sqrt{4\, m13\, m31 - m22^2}};$$

```
ExpCoefficientsToMatrix[expr_, x_, p_] :=
  CoefficientList[Exponent[expr, e], {x, p}, {3, 3}];

GaussIntR1[expr_, x_] := Module[{a, b, c, d, expo},
   expo = Exponent[expr, e];
   d = Coefficient[expr, e, expo];
   {c, b, a} = CoefficientList[expo, x, 3];
```
$$\lim_{\{aL,bL,cL\}\to\{a,b,c\}} \left(\frac{d\, e^{-\frac{bL^2}{4\, aL}+cL}\, \sqrt{\pi}}{\sqrt{-aL}}\right)$$
```
  ];

GaussIntR2[expr_, var1_, var2_] :=
  Module[{expo, coef, M, m11, m12, m13, m21, m22, m23, m31, m32,
    m33, m11L, m12L, m13L, m21L, m22L, m23L, m31L, m32L, m33L},
   expo = Exponent[expr, e];
   coef = Coefficient[expr, e, expo];
   {{m11, m12, m13}, {m21, m22, m23}, {m31, m32, m33}} =
    CoefficientList[expo, {var1, var2}, {3, 3}];
```
$$\lim_{\{m11L,m12L,m13L,m21L,m22L,m23L,m31L,m32L,m33L\}\to\{m11,m12,m13,m21,m22,m23,m31,m32,m33\}} (coef$$
```
      GaussianIntegralMatrix[
       {{m11L, m12L, m13L}, {m21L, m22L, m23L}, {m31L, m32L, m33L}}])
  ];
```

(* Trunated Gaussian integrals: $\int_{-\Delta/2}^{\Delta/2} e^{(c\ x^2+b\ x\ +a)}\,dx$ *)
```
ClearAll[TruncatedGaussInt];
TruncatedGaussInt[expr_, x_, Δ_] := Module[{a, b, c, d, expo, coef},
   expo = Exponent[expr, e];
   d = Coefficient[expr, e, expo];
   {c, b, a} = CoefficientList[expo, {x}, {3}];
```
$$\mathrm{Simplify}\left[d\, \frac{e^{-\frac{b^2}{4a}+c}\, \sqrt{\pi}\, \left(-\mathrm{Erfi}\left[\frac{b-a\,\Delta}{2\sqrt{a}}\right] + \mathrm{Erfi}\left[\frac{b+a\,\Delta}{2\sqrt{a}}\right]\right)}{2\sqrt{a}}\right]];$$

$$\mathrm{ResolvePD}[n\_, m\_, A\_, B\_, expr\_] := \lim_{A\to 0}\left(\partial_{\{A,\mathrm{Min}[m,n]\}}\left(\lim_{B\to 0}\left(\partial_{\{B,\mathrm{Max}[m,n]\}}(expr)\right)\right)\right);$$
```
ResolvePD2[a_] := ResolvePD[a[[2, 2, 1]], a[[2, 2, 2]],
   A2, B2, ResolvePD[a[[2, 1, 1]], a[[2, 1, 2]], A1, B1, a[[1]]]];
```

# Wigner of $\left|n\right\rangle\left\langle m\right|$

In[29]:=
```
ℏ = 1;
(* Wigner of |n⟩⟨m|. Ref: StackExchange answer
   to "What is the Winger of |n⟩⟨m|?" by Cosmas Zachos. *)
ClearAll[W, Walt];
```

$$W[n\_, m\_, q\_, p\_] := \lim_{x\to q}\left(\frac{(-1)^{Min[n,m]}}{\pi}\sqrt{\frac{(Min[n, m])!}{(Max[n, m])!}}\, e^{-(q^2+p^2)}\left(\sqrt{2}\,(x-(2\,UnitStep[n-m]-1)\,i\,p)\right)^{Abs[n-m]} LaguerreL\left[Min[n, m], Abs[n-m], 2\left(q^2+p^2\right)\right]\right);$$

$$Wg[\delta L\_, n\_, m\_, q\_, p\_] := Module\left[\{\delta, x\}, \lim_{\delta\to\delta L}\left(\lim_{x\to q}\left(\frac{(-1)^{Min[n,m]}}{\pi\,(1+2\,\delta)}\left(\frac{(1-2\,\delta)^{Min[n,m]}}{(1+2\,\delta)^{Max[n,m]}}\right)\sqrt{\frac{(Min[n, m])!}{(Max[n, m])!}}\, e^{\frac{-(q^2+p^2)}{1+2\,\delta}}\left(\sqrt{2}\,(x-(2\,UnitStep[n-m]-1)\,i\,p)\right)^{Abs[n-m]} LaguerreL\left[Min[n, m], Abs[n-m], \frac{2\left(q^2+p^2\right)}{1-4\,\delta^2}\right]\right)\right)\right];$$

$$Waltpd[A\_, B\_, n\_, m\_, q\_, p\_] := \frac{(-1)^{Min[n,m]}}{\pi\,\sqrt{n!\,m!}}\, e^{-(q^2+p^2)+A\,B-\sqrt{2}\,A\,(q+i\,p\,(-1+2\,UnitStep[n-m]))+\sqrt{2}\,B\,(q-i\,p\,(-1+2\,UnitStep[n-m]))};$$

$$Waltgpd[A\_, B\_, \delta\_, n\_, m\_, q\_, p\_] := \frac{(-1)^{Min[n,m]}}{\pi\,(1+2\,\delta)\,\sqrt{n!\,m!}}\left(\frac{1-2\,\delta}{1+2\,\delta}\right)^{Max[n,m]} e^{-\frac{(q^2+p^2)}{1+2\,\delta}+A\,B-A\,\frac{\sqrt{2}\,(q+i\,p\,(-1+2\,UnitStep[n-m]))}{1+2\,\delta}+B\,\frac{\sqrt{2}\,(q-i\,p\,(-1+2\,UnitStep[n-m]))}{1-2\,\delta}};$$

```
Walt[n_, m_, q_, p_] :=
  Module[{A, B}, ResolvePD[n, m, A, B, Waltpd[A, B, n, m, q, p]]];
```

(* gaussian filtered $W_{|n\rangle\langle m|}$ in the form of parametric differential of exponential *)

```
Waltg[δ_, n_, m_, q_, p_] :=
  Module[{A, B}, ResolvePD[n, m, A, B, Waltgpd[A, B, δ, n, m, q, p]]];
```

Associated Laguerre needs to be expressed in terms of Gaussians

For ordinary Laguerre polynomials we have :

$$\mathcal{L}_n(-z_1\,z_2) := \lim_{A\to 0}\left(\frac{1}{n!}\,\partial_{\{A,n\}}\left(\lim_{B\to 0}\left(\partial_{\{B,n\}}\left(e^{A\,B+A\,z_1+B\,z_2}\right)\right)\right)\right)$$

$$\mathcal{L}_n^{\alpha}(-z_1\,z_2) := \lim_{A\to 0}\left(\frac{z_2^{-\alpha}}{n!}\,\partial_{\{A,n\}}\left(\lim_{B\to 0}\left(\partial_{\{B,n+\alpha\}}\left(e^{AB+A\,z_1+B\,z_2}\right)\right)\right)\right)$$

$$\sum_{n=0}^{K}\sum_{m=0}^{K} \lim_{A\to 0}\left(\frac{1}{(n<)!}\,\partial_{\{A,n<\}}\left(\lim_{B\to 0}\left(\partial_{\{B,n>\}}\left(\text{remains}\right.\right.\right.\right.$$

until the end when we are tracing stuff in mode 2.

## *OLD code*
## Ideal k-photon subtraction

## *OLD code*
## THRESHOLD DETECTOR

## *OLD code*
## DYNE DETECTOR

## Ideal binomial code

## Two - mode beamsplitter output

In[85]:=

```
(* Ideal before beamsplitter *)
ClearAll[Wbinpm, Wbin01];
Wbinpm[μ_, 𝒩_, K_, r_, q_, p_] :=
```

$$\sum_{n=0}^{K}\sum_{m=0}^{K}\left(\frac{1}{2^{K}}\sqrt{\text{Binomial}[K, n]\ \text{Binomial}[K, m]}\ (-1)^{\mu n+\mu m}\, W[n\,\mathcal{N}, m\,\mathcal{N}, q, p]\right);$$

```
Wbin01[μ_, 𝒩_, K_, r_, q_, p_] :=
```

$$\sum_{n=0}^{\left\lfloor\frac{K-\mu}{2}\right\rfloor}\sum_{m=0}^{\left\lfloor\frac{K-\mu}{2}\right\rfloor}\left(\frac{1}{2^{K-1}}\sqrt{\text{Binomial}[K, 2n+\mu]\ \text{Binomial}[K, 2m+\mu]}\ W[(2n+\mu)\,\mathcal{N}, (2m+\mu)\,\mathcal{N}, q, p]\right);$$

```
(* Combining both into a single function: *)
Wbin[μ_, s_, 𝒩_, K_, r_, q_, p_] :=
```

$$\frac{1}{2^{s}}\sum_{n=0}^{\left\lfloor\frac{K-\mu}{2}\right\rfloor}\sum_{m=0}^{\left\lfloor\frac{K-\mu}{2}\right\rfloor}\left(\frac{1}{2^{K-1}}\sqrt{\text{Binomial}[K, 2n+\mu]\ \text{Binomial}[K, 2m+\mu]}\ W[(2n+\mu)\,\mathcal{N}, (2m+\mu)\,\mathcal{N}, q, p]\right) +$$

$$\frac{s}{2^{s}}\sum_{n=0}^{\left\lfloor\frac{K-1+\mu}{2}\right\rfloor}\sum_{m=0}^{\left\lfloor\frac{K-1+\mu}{2}\right\rfloor}\left(\frac{1}{2^{K-1}}\sqrt{\text{Binomial}[K, 2n+1-\mu]\ \text{Binomial}[K, 2m+1-\mu]}\ W[(2n+1-\mu)\,\mathcal{N}, (2m+1-\mu)\,\mathcal{N}, q, p]\right) +$$

$$\frac{(-1)^{s\mu}\, s}{2^{s}}\sum_{n=0}^{\left\lfloor\frac{K-\mu}{2}\right\rfloor}\sum_{m=0}^{\left\lfloor\frac{K+1-\mu}{2}\right\rfloor}\left(\frac{1}{2^{K-1}}\sqrt{\text{Binomial}[K, 2n+\mu]\ \text{Binomial}[K, 2m+1-\mu]}\ W[(2n+\mu)\,\mathcal{N}, (2m+1-\mu)\,\mathcal{N}, q, p]\right) +$$

$$\frac{(-1)^{s\mu}\, s}{2^{s}}\sum_{n=0}^{\left\lfloor\frac{K-1+\mu}{2}\right\rfloor}\sum_{m=0}^{\left\lfloor\frac{K-\mu}{2}\right\rfloor}\left(\frac{1}{2^{K-1}}\sqrt{\text{Binomial}[K, 2n+1-\mu]\ \text{Binomial}[K, 2m+\mu]}\ W[(2n+1-\mu)\,\mathcal{N}, (2m+\mu)\,\mathcal{N}, q, p]\right);$$

In[89]:=
```
ClearAll[Wbspm, Wbs01];
(* Just after beamsplitter *)
Wbspm[μ_, 𝒩_, K_, r_, q1_, p1_, q2_, p2_] :=
```
$$\sum_{n=0}^{K}\sum_{m=0}^{K}\sum_{p=0}^{n\mathcal{N}}\sum_{q=0}^{m\mathcal{N}}\left(\frac{1}{2^{K}}\sqrt{\text{Binomial[K, n] Binomial[K, m]}}\,(-1)^{\mu n+\mu m}\sqrt{\text{Binomial[n }\mathcal{N}\text{, p] Binomial[m }\mathcal{N}\text{, q]}}\left(\sqrt{1-r^{2}}\right)^{n\mathcal{N}-p+m\mathcal{N}-q} r^{p+q}\,\text{W[n }\mathcal{N}-\text{p, m }\mathcal{N}-\text{q, q1, p1]}\times\text{W[p, q, q2, p2]}\right);$$

```
Wbs01[μ_, 𝒩_, K_, r_, q1_, p1_, q2_, p2_] :=
```
$$\sum_{n=0}^{\lfloor\frac{K-\mu}{2}\rfloor}\sum_{m=0}^{\lfloor\frac{K-\mu}{2}\rfloor}\sum_{p=0}^{(2n+\mu)\mathcal{N}}\sum_{q=0}^{(2m+\mu)\mathcal{N}}\left(\frac{1}{2^{K-1}}\sqrt{\text{Binomial[K, 2 n + }\mu\text{] Binomial[K, 2 m + }\mu\text{]}}\sqrt{\text{Binomial[(2 n + }\mu\text{) }\mathcal{N}\text{, p] Binomial[(2 m + }\mu\text{) }\mathcal{N}\text{, q]}}\left(\sqrt{1-r^{2}}\right)^{(2n+\mu)\mathcal{N}-p+(2m+\mu)\mathcal{N}-q} r^{p+q}\,\text{W[(2 n + }\mu\text{) }\mathcal{N}-\text{p, (2 m + }\mu\text{) }\mathcal{N}-\text{q, q1, p1]}\times\text{W[p, q, q2, p2]}\right);$$

```
ClearAll[Wgpm, Wg01];
(* after gaussian filtering mode 2 *)
Wgpm[μ_, 𝒩_, K_, r_, δ_, q1_, p1_, q2_, p2_] :=
```
$$\sum_{n=0}^{K}\sum_{m=0}^{K}\sum_{p=0}^{n\mathcal{N}}\sum_{q=0}^{m\mathcal{N}}\left(\frac{1}{2^{K}}\sqrt{\text{Binomial[K, n] Binomial[K, m]}}\,(-1)^{\mu n+\mu m}\sqrt{\text{Binomial[n }\mathcal{N}\text{, p] Binomial[m }\mathcal{N}\text{, q]}}\,\text{W[n }\mathcal{N}-\text{p, m }\mathcal{N}-\text{q, q1, p1]}\left(\sqrt{1-r^{2}}\right)^{n\mathcal{N}-p+m\mathcal{N}-q} r^{p+q}\lim_{\delta L\to\delta^{+}}(\text{Wg[}\delta\text{L, p, q, q2, p2]})\right);$$

```
Wg01[μ_, 𝒩_, K_, r_, δ_, q1_, p1_, q2_, p2_] :=
```
$$\sum_{n=0}^{\lfloor\frac{K-\mu}{2}\rfloor}\sum_{m=0}^{\lfloor\frac{K-\mu}{2}\rfloor}\sum_{p=0}^{(2n+\mu)\mathcal{N}}\sum_{q=0}^{(2m+\mu)\mathcal{N}}\left(\frac{1}{2^{K-1}}\sqrt{\text{Binomial[K, 2 n + }\mu\text{] Binomial[K, 2 m + }\mu\text{]}}\sqrt{\text{Binomial[(2 n + }\mu\text{) }\mathcal{N}\text{, p] Binomial[(2 m + }\mu\text{) }\mathcal{N}\text{, q]}}\left(\sqrt{1-r^{2}}\right)^{(2n+\mu)\mathcal{N}-p+(2m+\mu)\mathcal{N}-q} r^{p+q}\,\text{W[(2 n + }\mu\text{) }\mathcal{N}-\text{p, (2 m + }\mu\text{) }\mathcal{N}-\text{q, q1, p1]}\lim_{\delta L\to\delta^{+}}(\text{Wg[}\delta\text{, p, q, q2, p2]})\right);$$

In[95]:=
```
(* For computational speedup,
the individual terms in the sum should be computed and stored *)
ClearAll[WaltgpmTablepd, Waltg01Tablepd];
WaltgpmTablepd[A_, B_, C_, D_, μ_, 𝒩_, K_, δ_, r_] := Table[Table[
    {1/2^K Sqrt[Binomial[K, n] Binomial[K, m]] (-1)^(μ n+μ m)
      Sqrt[Binomial[n 𝒩, p] Binomial[m 𝒩, q]] Waltpd[A, B, n 𝒩 - p, m 𝒩 - q, q1, p1]
      (Sqrt[1 - r^2])^(n 𝒩-p+m 𝒩-q) r^(p+q) lim_{δL→δ⁺} (Waltgpd[C, D, δL, p, q, q2, p2])
     , {{n 𝒩 - p, m 𝒩 - q}, {p, q}}}
    , {p, 0, n 𝒩}, {q, 0, m 𝒩}], {n, 0, K}, {m, 0, K}];
Waltg01Tablepd[A_, B_, C_, D_, μ_, 𝒩_, K_, δ_, r_] := Table[Table[
    {1/2^(K-1) Sqrt[Binomial[K, 2 n + μ] Binomial[K, 2 m + μ]]
      Sqrt[Binomial[(2 n + μ) 𝒩, p] Binomial[(2 m + μ) 𝒩, q]] (Sqrt[1 - r^2])^((2 n +μ) 𝒩-p+(2 m+μ) 𝒩-q)
      r^(p+q) Waltpd[A, B, (2 n + μ) 𝒩 - p, (2 m + μ) 𝒩 - q, q1, p1]
      lim_{δL→δ⁺} (Waltgpd[C, D, δ, p, q, q2, p2]),
     {{(2 n + μ) 𝒩 - p, (2 m + μ) 𝒩 - q}, {p, q}}}
    , {p, 0, (2 n + μ) 𝒩}, {q, 0, (2 m + μ) 𝒩}], {n, 0, Floor[(K - μ)/2]}, {m, 0, Floor[(K - μ)/2]}];
```

# THRESHOLD DETECTOR

In[98]:=
```
ClearAll[PsTHRpm, PsTHR01, WunTHRpm, WunTHR01, WaltunTHRpmTablepd,
  WaltunTHR01Tablepd, overlapaltTHRpmTablepd, overlapaltTHR01Tablepd,
  fidelityZPSTHRpm, fidelityZPSTHR01, purityTHRpm, purityTHR01 ];
(* Succcess probability *)
PsTHRpm[μ_, N_, K_, r_, η_] := Sum_{k=0}^{K N} 1/2^K Binomial[K, (k)]
   Sum_{m=0}^{(2 k +1) N} Binomial[(k) N, m] lim_{rL→r} ((Sqrt[1 - rL^2])^(2 (k) N-2 m) rL^(2 m)) lim_{ηL→η} ((1 - ηL)^m);

PsTHR01[μ_, N_, K_, r_, η_] := Sum_{k=0}^{(2 Floor[(K-μ)/2]+μ) N} 1/2^(K-1) Binomial[K, (2 k + μ)] Sum_{m=0}^{(2 k +μ) N} Binomial[
    (2 k + μ) N, m] lim_{rL→r} ((Sqrt[1 - rL^2])^(2 (2 k +μ) N-2 m) rL^(2 m)) lim_{ηL→η} ((1 - ηL)^m);
(* Unnormalized states after THR heralding *)
WunTHRpm[μ_, 𝒩_, K_, r_, η_, q_, p_] :=
  Sum_{n=0}^{K} Sum_{m=0}^{K} (1/2^K Sqrt[Binomial[K, n] Binomial[K, m]] (-1)^(μ n+μ m)
    Sum_{k=0}^{Min[n,m] 𝒩} Sqrt[Binomial[n 𝒩, k] Binomial[m 𝒩, k]] (Sqrt[1 - r^2])^(n 𝒩+m 𝒩-2 k)
```

```
        r^(2 k) lim_(ηL→η) ((1 - ηL)^k) W[n 𝒩 - k, m 𝒩 - k, q, p]);
WunTHR01[μ_, 𝒩_, K_, r_, η_, q_, p_] :=
  Sum[Sum[(1/2^(K-1) Sqrt[Binomial[K, 2 n + μ] Binomial[K, 2 m + μ]]
       Sum[Sqrt[Binomial[(2 n + μ) 𝒩, k] Binomial[(2 m + μ) 𝒩, k]]
        (Sqrt[1 - r^2])^((2 n + μ) 𝒩 + (2 m + μ) 𝒩 - 2 k) r^(2 k)
        lim_(ηL→η) ((1 - ηL)^k) W[(2 n + μ) 𝒩 - k, (2 m + μ) 𝒩 - k, q, p],
        {k, 0, (2 Min[n, m] + μ) 𝒩}]),
     {m, 0, Floor[(K - μ)/2]}], {n, 0, Floor[(K - μ)/2]}];

(* For computational speedup,
the individual terms in the sum should be computed and stored *)
WaltunTHRpmTablepd[A_, B_, μ_, 𝒩_, K_, r_, η_, q_, p_] := Module[{a},
   a = {};
   Table[Table[
     AppendTo[a,
      {1/2^K Sqrt[Binomial[K, n] Binomial[K, m]] (-1)^(μ n + μ m)
        Sqrt[Binomial[n 𝒩, k] Binomial[m 𝒩, k]] (Sqrt[1 - r^2])^(n 𝒩 + m 𝒩 - 2 k)
        r^(2 k) lim_(ηL→η) ((1 - ηL)^k) Waltpd[A, B, n 𝒩 - k, m 𝒩 - k, q, p]
       , {n 𝒩 - k, m 𝒩 - k
         (* We also need to append the Fock basis eigenvalues
          for the input to ResolvePD function at the end *)
         , n 𝒩, m 𝒩
         (* The last two numbers are in case of debugging
          and don't serve any other computational benefit *)}}
     ]
    , {k, 0, Min[n, m] 𝒩}], {n, 0, K}, {m, 0, K}];
   a];
WaltunTHR01Tablepd[A_, B_, μ_, 𝒩_, K_, r_, η_, q_, p_] :=
  Module[{a},
   a = {};
   Table[Table[
     AppendTo[a,
      {1/2^(K-1) Sqrt[Binomial[K, 2 n + μ] Binomial[K, 2 m + μ]]
        Sqrt[Binomial[(2 n + μ) 𝒩, k] Binomial[(2 m + μ) 𝒩, k]]
```

```
        (√(1 - r^2))^((2 n + μ) 𝒩 + (2 m + μ) 𝒩 - 2 k) r^(2 k)
        lim_{ηL→η} ((1 - ηL)^k) Waltpd[A, B, (2 n + μ) 𝒩 - k, (2 m + μ) 𝒩 - k, q, p]
      , {(2 n + μ) 𝒩 - k, (2 m + μ) 𝒩 - k
        (* We also need to append the Fock basis eigenvalues
         for the input to ResolvePD function at the end *)
        , (2 n + μ) 𝒩, (2 m + μ) 𝒩
        (* The last two numbers are in case of debugging
         and don't serve any other computational benefit *)}}
    ]
    , {k, 0, (2 Min[n, m] + μ) 𝒩}], {n, 0, ⌊(K - μ)/2⌋}, {m, 0, ⌊(K - μ)/2⌋}];
  a];

(* Overlap between unnormalized Wigners for two THR outputs *)
overlapaltTHRpmTablepd[μ_, 𝒩_, K_, r_, η1_, η2_] :=
  Module[{a, b, res, q, p},
   a = WaltunTHRpmTablepd[A1, B1, μ, 𝒩, K, r, η1, q, p];
   b = WaltunTHRpmTablepd[A2, B2, μ, 𝒩, K, r, η2, q, p];
   res = Table[
     {2 π GaussIntR2[a[[i]][[1]] × b[[j]][[1]], q, p], {a[[i]][[2]], b[[j]][[2]]}}
     , {i, Dimensions[a][[1]]}, {j, 1, Dimensions[b][[1]]}]
  ];
overlapaltTHR01Tablepd[μ_, 𝒩_, K_, r_, η1_, η2_] :=
  Module[{a, b, res, q, p},
   a = WaltunTHR01Tablepd[A1, B1, μ, 𝒩, K, r, η1, q, p];
   b = WaltunTHR01Tablepd[A2, B2, μ, 𝒩, K, r, η2, q, p];
   res = Table[
     {2 π GaussIntR2[a[[i]][[1]] × b[[j]][[1]], q, p], {a[[i]][[2]], b[[j]][[2]]}}
     , {i, Dimensions[a][[1]]}, {j, 1, Dimensions[b][[1]]}]
  ];

(* Fidelity b/w THR and ZPS outputs *)
fidelityZPSTHRpm[μ_, 𝒩_, K_, r_, η_] :=
  Module[{o},
   o = overlapaltTHRpmTablepd[μ, 𝒩, K, r, η1, η2] /. {η1 → η, η2 → 1};
    Total[Map[ResolvePD2, o, {2}], 2] / (PsTHRpm[μ, 𝒩, K, r, η] × PsTHRpm[μ, 𝒩, K, r, 1])];
fidelityZPSTHR01[μ_, 𝒩_, K_, r_, η_] :=
  Module[{o},
   o = overlapaltTHR01Tablepd[μ, 𝒩, K, r, η1, η2] /. {η1 → η, η2 → 1};
    Total[Map[ResolvePD2, o, {2}], 2] / (PsTHR01[μ, 𝒩, K, r, η] × PsTHR01[μ, 𝒩, K, r, 1])];

(* Purity of THR output *)
```

```
purityTHRpm[μ_, 𝓝_, K_, r_, η_] :=
  Total[Map[ResolvePD2, overlapaltTHRpmTablepd[μ, 𝓝, K, r, η, η], {2}], 2] /
   PsTHRpm[μ, 𝓝, K, r, η]^2;
purityTHR01[μ_, 𝓝_, K_, r_, η_] :=
  Total[Map[ResolvePD2, overlapaltTHR01Tablepd[μ, 𝓝, K, r, η, η], {2}], 2] /
   PsTHR01[μ, 𝓝, K, r, η]^2;
```

# HOMODYNE DETECTOR

In[337]:=

```
δHOM[η_] := (1 - η)/(2 η);
ClearAll[WunHOM01pd, WunHOMpmpd, WgHOM];
 (* Leave this mode 2 part undefined until just before
  computation of the final output or plug in values for δ,
 Δ before using this module so that it doesn't hurt GaussInts *);
WunHOMpmpd[A_, B_, μ_, 𝒩_, K_, r_, δ_, Δ_, θ_, q1_, p1_] :=
  Module[{a},
   a = {};
   Table[Table[
     AppendTo[a,
      {1/2^K Sqrt[Binomial[K, n] Binomial[K, m]]
        (-1)^(μ n+μ m) Sqrt[Binomial[n 𝒩, p] Binomial[m 𝒩, q]]
        Waltpd[A, B, n 𝒩 - p, m 𝒩 - q, q1, p1] (Sqrt[1 - r^2])^(n 𝒩-p+m 𝒩-q) r^(p+q)
        (WgHOM[δ, p, q, Δ, θ](* Leave this mode 2 part undefined until just
          before computation of the final output or plug in values for δ,
         Δ before using this module so that it doesn't hurt GaussInts *))
       , {n 𝒩 - p, m 𝒩 - q, n 𝒩, m 𝒩}}
     ]
     , {p, 0, n 𝒩}, {q, 0, m 𝒩}], {n, 0, K}, {m, 0, K}];
   a];
WunHOM01pd[A_, B_, μ_, 𝒩_, K_, r_, δ_, Δ_, θ_, q1_, p1_] :=
  Module[{a},
   a = {};
   Table[Table[
     AppendTo[a,
      {1/2^(K-1) Sqrt[Binomial[K, 2 n + μ] Binomial[K, 2 m + μ]]
        Sqrt[Binomial[(2 n + μ) 𝒩, p] Binomial[(2 m + μ) 𝒩, q]]
        (Sqrt[1 - r^2])^((2 n +μ) 𝒩-p+(2 m+μ) 𝒩-q) r^(p+q) Waltpd[A, B, (2 n + μ) 𝒩 - p, (2 m + μ) 𝒩 - q, q1,
         p1] (WgHOM[δ, p, q, Δ, θ](* Leave this WgHOM function undefined until
          just before computation of the final output or plug in values for δ,
         Δ before using this module so that it doesn't hurt GaussInts *))
       , {(2 n + μ) 𝒩 - p, (2 m + μ) 𝒩 - q, n 𝒩, m 𝒩}}
     ]
     , {p, 0, (2 n + μ) 𝒩}, {q, 0, (2 m + μ) 𝒩}], {n, 0, Floor[(K - μ)/2]}, {m, 0, Floor[(K - μ)/2]}];
   a];
```

```
(* Choosing 𝒩=2, K=3 *)
```

In[•]:=
```
ClearAll[ArrayOfWunHOMpm0Terms];
ArrayOfWunHOMpm0Terms[A_, B_, r_, δ_, Δ_, θ_, q1_, p1_] =
  Simplify[WunHOMpmpd[A, B, 0 , 2, 3, r, δ, Δ, θ, q1, p1]];
```

In[•]:=
```
ClearAll[ArrayOfWunHOMpm1Terms];
ArrayOfWunHOMpm1Terms[A_, B_, r_, δ_, Δ_, θ_, q1_, p1_] =
  Simplify[WunHOMpmpd[A, B, 1 , 2, 3, r, δ, Δ, θ, q1, p1]];
```

In[•]:=
```
ClearAll[ArrayOfWunHOM010Terms];
ArrayOfWunHOM010Terms[A_, B_, r_, δ_, Δ_, θ_, q1_, p1_] =
  Simplify[WunHOM01pd[A, B, 0, 2, 3, r, δ, Δ, θ, q1, p1]];
```

In[•]:=
```
ClearAll[ArrayOfWunHOM011Terms];
ArrayOfWunHOM011Terms[A_, B_, r_, δ_, Δ_, θ_, q1_, p1_] =
 Simplify[WunHOM01pd[A, B, 1, 2, 3, r, δ, Δ, θ, q1, p1]];
```

In[•]:=
```
(*testing...*)ArrayOfWunHOMpm0Terms[A, B, r, δ, Δ, θ, q1, p1][[1 ;; 4]]
```

Out[•]=

$$\left\{\left\{\frac{e^{A\,B-p1^2-q1^2+\sqrt{2}\,B\,(-i\,p1+q1)-\sqrt{2}\,A\,(i\,p1+q1)}\,\text{WgHOM}[\delta,0,0,\Delta,\theta]}{8\,\pi},\{0,0,0,0\}\right\},\right.$$

$$\left\{-\frac{\sqrt{\frac{3}{2}}\,e^{A\,\left(B+i\,\sqrt{2}\,(p1+i\,q1)\right)+\left(\sqrt{2}\,B+i\,p1-q1\right)\,(i\,p1+q1)}\,\left(-1+r^2\right)\,\text{WgHOM}[\delta,0,0,\Delta,\theta]}{8\,\pi},\{0,2,0,2\}\right\},$$

$$\left\{\frac{\sqrt{\frac{3}{2}}\,e^{A\,\left(B+i\,\sqrt{2}\,(p1+i\,q1)\right)+\left(\sqrt{2}\,B+i\,p1-q1\right)\,(i\,p1+q1)}\,r\,\sqrt{1-r^2}\,\text{WgHOM}[\delta,0,1,\Delta,\theta]}{4\,\pi},\{0,1,0,2\}\right\},$$

$$\left.\left\{\frac{\sqrt{3}\,e^{A\,B-p1^2-q1^2+\sqrt{2}\,B\,(-i\,p1+q1)-\sqrt{2}\,A\,(i\,p1+q1)}\,r^2\,\text{WgHOM}[\delta,0,2,\Delta,\theta]}{8\,\pi},\{0,0,0,2\}\right\}\right\}$$

## $\Pi_{q_2}^{(\theta)}$ on $\left|+_L\right\rangle$

In[•]:=
```
ClearAll[WgHOM];
(* Define WgHOM before defining this and clear the function again *)
WunHOMpm0[r_, δ_, Δ_, θ_, q1_, p1_] = Module[{tmp, res, A, B},
   tmp = ArrayOfWunHOMpm0Terms[A, B, r, δ, Δ, θ, q1, p1];
   res = Total[Table[
      ResolvePD[tmp[[i, 2, 1]], tmp[[i, 2, 2]], A, B, tmp[[i, 1]]]
      , {i, 1, Dimensions[tmp][[1]]}]];
   res];
```

In[ ]:=
```
ClearAll[WgHOM];
(* Define WgHOM AFTER defining this and clear the function again until ready
 to compute *)PsHOMpm0[r_, δ_, Δ_, θ_] = Module[{tmp, res, A, B, q1, p1},
  tmp  = ArrayOfWunHOMpm0Terms[A, B, r, δ, Δ, θ, q1, p1];
  res  = Total[Table[
      ResolvePD[tmp⟦i, 2, 1⟧, tmp⟦i, 2, 2⟧, A, B, GaussIntR2[tmp⟦i, 1⟧, q1, p1]]
      , {i, 1, Dimensions[tmp]⟦1⟧}]];
  res];
```

In[ ]:=
```
ClearAll[WgHOM];(* Define WgHOM AFTER defining this
 and clear the function again until ready to compute *)
fidelityunZPSHOMpm0[r_, δ_, Δ_, θ_] = Module[{tmp1, tmp2, res, q1, p1},
   tmp1 = ArrayOfWunHOMpm0Terms[A1, B1, r, δ, Δ, θ, q1, p1];
   tmp2 = WaltunTHRpmTablepd[A2, B2, 0, 2, 3, r, 1, q1, p1];
   res  = Total[Table[ResolvePD2[
       {2 π GaussIntR2[tmp1⟦i, 1⟧ × tmp2⟦j, 1⟧, q1, p1], {tmp1⟦i, 2⟧, tmp2⟦j, 2⟧}}],
      {i, 1, Dimensions[tmp1]⟦1⟧}, {j, 1, Dimensions[tmp2]⟦1⟧}], 2];
   res];
```

In[383]:=
```
ClearAll[fidelityZPSHOMpm0];
fidelityZPSHOMpm0[r_, δ_, Δ_, θ_] := fidelityunZPSHOMpm0[r, δ, Δ, θ] / (PsTHRpm[0, 2, 3, r, 1] × PsHOMpm0[r, δ, Δ, θ]);

(*ClearAll[WgHOM];
(* Define WgHOM AFTER defining this and
 clear the function again until ready to compute *)
purityunHOMpm0[r_,δ_,Δ_,θ_]=Module[{tmp1,tmp2,res,q1,p1},
  tmp1= ArrayOfWunHOMpm0Terms[A1,B1,r,δ,Δ,θ,q1,p1];
  tmp2=ArrayOfWunHOMpm0Terms[A2,B2,r,δ,Δ,θ,q1,p1];
  res =Total[Table[ResolvePD2[
      {2π GaussIntR2[tmp1⟦i⟧⟦1⟧tmp2⟦j⟧⟦1⟧,q1,p1],{tmp1⟦i,2⟧,tmp2⟦j,2⟧}}],
     {i,1,Dimensions[tmp1]⟦1⟧},{j,1,Dimensions[tmp2]⟦1⟧}],2];
  res];*)

(*purityHOMpm0[r_,δ_,Δ_]=purityunHOMpm0[r,δ,Δ]/PsHOMpm0[r,δ,Δ]^2;*)
```

$\Pi_{q_2^{(\theta)}}$ on $\left|-_L\right\rangle$

$\Pi_{q_2^{(\theta)}}$ on $\left|0_L\right\rangle$

$\Pi_{q_2^{(\theta)}}$ on $\left|1_L\right\rangle$

# Compiling $W_{\text{un}}$, Ps, $\mathcal{F}$ for HOM

```
In[ ]:= (* HOM *)
ClearAll[WgHOM];
WgHOMintPpd[A_, B_, δ_, n_, m_, q_, θ_] := Module[{p}, Simplify[GaussIntR1[
     Waltgpd[A, B, δ, n, m, q Cos[θ] - p Sin[θ], p Cos[θ] + q Sin[θ]], p], δ > 0]];
WgHOMintPQpd[A_, B_, δ_, n_, m_, Δ_, θ_] := Module[{q},
   Simplify[TruncatedGaussInt[WgHOMintPpd[A, B, δ, n, m, q, θ], q, Δ], δ > 0]];
WgHOMModule[δ_, n_, m_, Δ_, θ_] := Module[{A, B, δL},
    lim_{δL→δ} (Simplify[ResolvePD[n, m, A, B, WgHOMintPQpd[A, B, δL, n, m, Δ, θ]]])];

In[ ]:= Table[WgHOM[δ_, n, m, Δ_, θ_] = WgHOMModule[δ, n, m, Δ, θ], {n, 0, 6}, {m, 0, 6}]

In[3]:= (* If table backup is avaialble use this after running that cell *)
Table[WgHOM[δ_, n, m, Δ_, θ_] = WgHOMTableBackup[δ, Δ, θ][[n + 1, m + 1]],
  {n, 0, 6}, {m, 0, 6}];
```

### Table Backup

# [Rslt] $W_{\text{un}}$, Ps, & $\mathcal{F}_{\text{un}}$ for $\Pi_{q_2^{(\theta)}} \left| +_L \right\rangle$

### $W_{\text{un}}$

### Ps

### $\mathcal{F}_{\text{un}}$

### [Plt] $\Delta\mathcal{F}$ & $\Delta$Ps @ $\theta = 0$

### [Plt] $W$ @ $\theta$=0

## [Plt] ΔF & ΔPs @ θ = π/2

In[420]:=

```
ClearAll[ΔF, ΔPs];
ΔFpm0[r_, η_, δ_, Δ_, θ_] :=
  Re[fidelityZPSHOMpm0[r, δ, Δ, θ] - FidelityTHR[(*j*)1, 2, 3, r, η]];
ΔPspm0[r_, η_, δ_, Δ_, θ_] := Re[PsHOMpm0[r, δ, Δ, θ] - PsTHRpm[0, 2, 3, r, η]];
ClearAll[r, θ, ηH, Δlimit, plotpoints, Δ, δ, plotROHOM, plotRIHOM];
r = √.5;
"Dyne efficiency chosen:"
ηH = .85
plotpoints = 60;
Δlimit = 7;
δ = δHOM[ηH];
θ = π / 2;
cols =
  Reverse[RGBColor /@ {(*"#053061","#2166ac",*)"#4393c3", "#92c5de", "#d1e5f0",
     "#f7f7f7", "#fddbc7", "#f4a582", "#d6604d"(*,"#b2182b","#67001f"*)}];
"HOMODYNE:"
texta = Text[Style["(a)", {FontSize → Large, FontFamily → "Arial", Black, Bold}],
   {4.5/5 Δlimit, .9}, {0, 0}];
txta = Graphics[{texta}];
plotROHOM = DensityPlot[HeavisideTheta[ΔFpm0[r, η, δ, Δ, θ]],
   {Δ, 0, Δlimit}, {η, 0, 1}, PlotPoints → plotpoints,
   ColorFunction → (Blend[cols, #] &), ColorFunctionScaling → True,
   PlotRange → All, FrameLabel → {"Δ (in units of √ℏ)", "η"}, LabelStyle →
    {FontSize → Large, FontFamily → "Arial", Black}, ImageSize → Medium];
plotRIHOM = DensityPlot[HeavisideTheta[ΔPspm0[r, η, δ, Δ, θ]],
   {Δ, 0, Δlimit}, {η, 0, 1}, PlotPoints → plotpoints,
   ColorFunction → (Blend[{RGBColor[0, 0, 0, 0.3], RGBColor[1, 1, 1, 0]}, #] &),
   ColorFunctionScaling → False, PlotRange → All,
   FrameLabel → {"Δ (in units of √ℏ)", "η"}, LabelStyle →
    {FontSize → Large, FontFamily → "Arial", Black}, ImageSize → Medium];
Show[{plotROHOM, plotRIHOM, txta}]
ClearAll[r, θ, ηH, Δlimit, plotpoints, Δ, δ, plotROHOM, plotRIHOM, txta, texta];
```

Out[425]=

```
Dyne efficiency chosen:
```

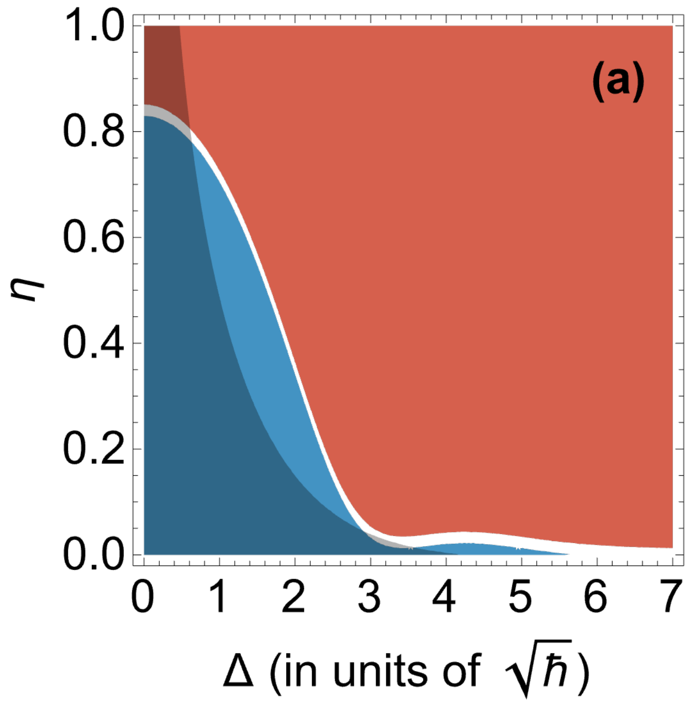
Out[426]=

```
0.85
```

Out[432]=

```
HOMODYNE:
```

Out[437]=



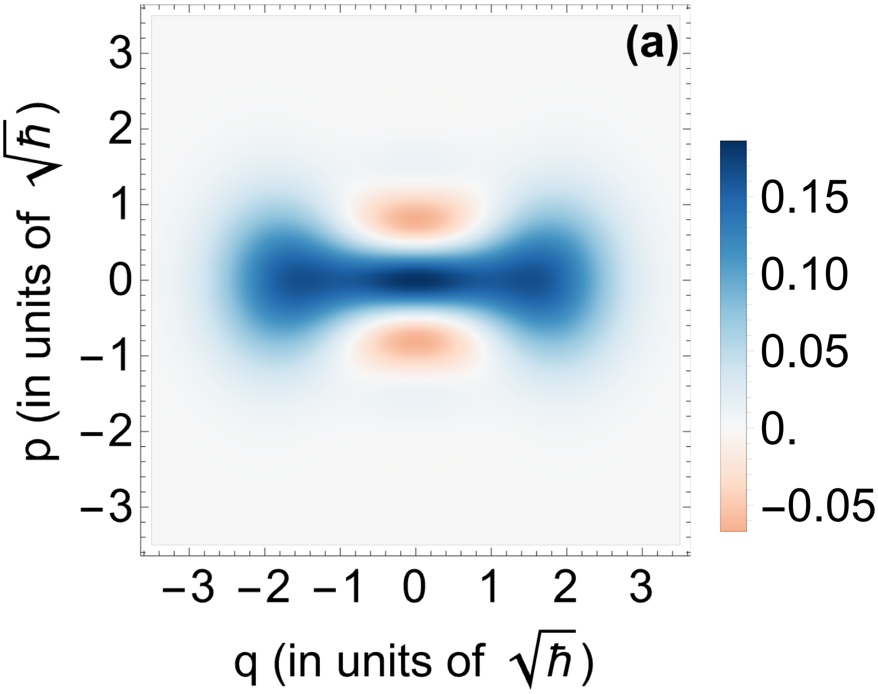

## [Plt] $W$ @ $\theta = \pi/2$

In[439]:=

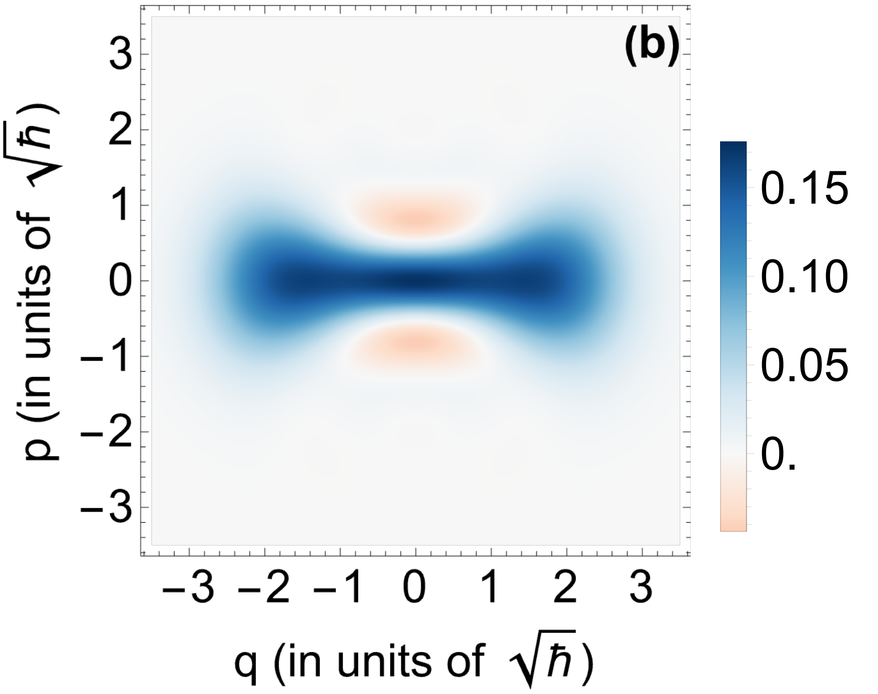
```
(* Options for the density plot *)
cols = RGBColor /@ {"#053061", "#2166ac", "#4393c3", "#92c5de", "#d1e5f0",
    "#f7f7f7", "#fddbc7", "#f4a582", "#d6604d", "#b2182b", "#67001f"};
limits = 3.5;
range = 50;
ClearAll[ηH, η, r, δ, Δ, θ];
ηH = 0.85;
η = .75;
r = √.5;
δ = δHOM[ηH];
Δ = .75;
θ = π / 2;

ClearAll[mat1, mat2];
(* Construct matrix to plot *)
mat1 = Table[Re[Chop[WunHOMpm0[r, δ, Δ, θ, (q - 1) (limits / range) - limits,
        (p - 1) (limits / range) - limits] / PsHOMpm0[r, δ, Δ, θ]]],
    {p, 1, 2 range + 1}, {q, 1, 2 range + 1}];
(* Normalize the mat *)
mat1 = mat1 / (1 (*Abs[ Total[mat1,2]] (limits/range)^2 *));
(* Construct matrix to plot *)
mat2 = Table[Re[Chop[WunTHRpm[0, 2, 3, r, η, (q - 1) (limits / range) - limits,
        (p - 1) (limits / range) - limits] / PsTHRpm[0, 2, 3, r, η]]],
```

```
   {p, 1, 2 range + 1}, {q, 1, 2 range + 1}];
(* Normalize the mat *)
mat2 = mat2 / (1(*Abs[ Total[mat2,2]] (limits/range)^2*));

(* Plotting *)
texta = Text[Style["(a)", {FontSize → Large, FontFamily → "Arial", Black, Bold}],
   {.9 limits, .9 limits}, {0, 0}];
txta = Graphics[{texta}];
plota = ListDensityPlot[mat1,
   DataRange → {{-limits, limits}, {-limits, limits}}, ColorFunction →
    (Blend[Reverse[cols], (# / Max[{Abs[Max[mat1]], Abs[Min[mat1]]}] + 1) / 2] &),
   ColorFunctionScaling → False, PlotRange → Full,
   FrameLabel → {"q (in units of √ℏ)", "p (in units of √ℏ)"},
   LabelStyle → {FontSize → Large, FontFamily → "Arial", Black},
   AxesStyle → {Thick, Black}, ImageSize → Medium,
   PlotLegends → Automatic, InterpolationOrder → 1];
Show[{plota, txta}]

textb = Text[Style["(b)", {FontSize → Large, FontFamily → "Arial", Black, Bold}],
   {.9 limits, .9 limits}, {0, 0}];
txtb = Graphics[{textb}];
plotb = ListDensityPlot[mat2,
   DataRange → {{-limits, limits}, {-limits, limits}}, ColorFunction →
    (Blend[Reverse[cols], (# / Max[{Abs[Max[mat2]], Abs[Min[mat2]]}] + 1) / 2] &),
   ColorFunctionScaling → False, PlotRange → Full,
   FrameLabel → {"q (in units of √ℏ)", "p (in units of √ℏ)"},
   LabelStyle → {FontSize → Large, FontFamily → "Arial", Black},
   AxesStyle → {Thick, Black}, ImageSize → Medium,
   PlotLegends → Automatic, InterpolationOrder → 1];
Show[{plotb, txtb}]

ClearAll[mat1, mat2];
ClearAll[ηH, η, r, δ, Δ, θ, plota, plotb, txta, txtb, texta, textb];
```

Out[457]=




Out[461]=




## [Rslt] $W_{\text{un}}$, Ps, & $\mathcal{F}_{\text{un}}$ for $\Pi_{q_2^{(\pi/4)}} \left| 0_L \right\rangle$

## *old*Compiled $W_{\text{un}}$, Ps, $\mathcal{F}_{\text{un}}$, and $\text{tr}\left(\rho^2\right)$ functions for $\Pi_{q_2^{(0)}} \left| +_L \right\rangle$

# *old*Compiled $W_{\mathrm{un}}$, Ps, $\mathcal{F}$, and $\mathrm{tr}\left(\rho^2\right)$ functions for $\Pi_{q_2^{(\pi/4)}}\left|+_L\right\rangle$

# HETERODYNE DETECTOR

In[132]:=

```
δHET[η_] := (2 - η)/(2 η);
ClearAll[WunHET01pd, WunHETpmpd, WgHET];
 (* Leave this mode 2 part undefined until just before
  computation of the final output or plug in values for δ,
 Δ before using this module so that it doesn't hurt GaussInts *);
WunHETpmpd[A_, B_, μ_, N_, K_, r_, δ_, Δq_, Δp_, q1_, p1_] :=
  Module[{a},
   a = {};
   Table[Table[
     AppendTo[a,
      {1/2^K Sqrt[Binomial[K, n] Binomial[K, m]]
        (-1)^(μ n+μ m) Sqrt[Binomial[n N, p] Binomial[m N, q]]
        Waltpd[A, B, n N - p, m N - q, q1, p1] (Sqrt[1 - r^2])^(n N-p+m N-q) r^(p+q)
        (WgHET[δ, p, q, Δq, Δp] (* Leave this mode 2 part undefined until just
          before computation of the final output or plug in values for δ,
         Δ before using this module so that it doesn't hurt GaussInts *))
       , {n N - p, m N - q, n N, m N}}
     ]
     , {p, 0, n N}, {q, 0, m N}], {n, 0, K}, {m, 0, K}];
   a];
WunHET01pd[A_, B_, μ_, N_, K_, r_, δ_, Δq_, Δp_, q1_, p1_] :=
  Module[{a},
   a = {};
   Table[Table[
     AppendTo[a,
      {1/2^(K-1) Sqrt[Binomial[K, 2 n + μ] Binomial[K, 2 m + μ]]
        Sqrt[Binomial[(2 n + μ) N, p] Binomial[(2 m + μ) N, q]]
        (Sqrt[1 - r^2])^((2 n +μ) N-p+(2 m+μ) N-q) r^(p+q)
        Waltpd[A, B, (2 n + μ) N - p, (2 m + μ) N - q, q1, p1]
        (WgHET[δ, p, q, Δq, Δp] (* Leave this WgHET function undefined until
          just before computation of the final output or plug in values for δ,
         Δ before using this module so that it doesn't hurt GaussInts *))
       , {(2 n + μ) N - p, (2 m + μ) N - q, n N, m N}}
     ]
     , {p, 0, (2 n + μ) N}, {q, 0, (2 m + μ) N}], {n, 0, Floor[(K - μ)/2]}, {m, 0, Floor[(K - μ)/2]}];
   a];
```

```
(* Choosing 𝒩=2, K=3 *)

In[•]:= ClearAll[ArrayOfWunHETpm0Terms];
ArrayOfWunHETpm0Terms[A_, B_, r_, δ_, Δq_, Δp_, q1_, p1_] =
  Simplify[WunHETpmpd[A, B, 0 , 2, 3, r, δ, Δq, Δp, q1, p1]];

In[•]:= ClearAll[ArrayOfWunHETpm1Terms];
ArrayOfWunHETpm1Terms[A_, B_, r_, δ_, Δq_, Δp_, q1_, p1_] =
  Simplify[WunHETpmpd[A, B, 1 , 2, 3, r, δ, Δq, Δp, q1, p1]];

In[•]:= ClearAll[ArrayOfWunHET010Terms];
ArrayOfWunHET010Terms[A_, B_, r_, δ_, Δq_, Δp_, q1_, p1_] =
  Simplify[WunHET01pd[A, B, 0, 2, 3, r, δ, Δq, Δp, q1, p1]];

In[•]:= ClearAll[ArrayOfWunHET011Terms];
ArrayOfWunHET011Terms[A_, B_, r_, δ_, Δq_, Δp_, q1_, p1_] =
 Simplify[WunHET01pd[A, B, 1, 2, 3, r, δ, Δq, Δp, q1, p1]];

In[•]:= (*testing...*)ArrayOfWunHETpm0Terms[A, B, r, δ, Δq, Δp, q1, p1][[1 ;; 4]]
Out[•]=
```

$$\left\{\left\{\frac{e^{A\,B-p1^2-q1^2+\sqrt{2}\,B\,(-i\,p1+q1)-\sqrt{2}\,A\,(i\,p1+q1)}\,\text{WgHET}[\delta,0,0,\Delta q,\Delta p]}{8\,\pi},\{0,0,0,0\}\right\},\right.$$

$$\left\{-\frac{\sqrt{\frac{3}{2}}\,e^{A\,\left(B+i\,\sqrt{2}\,(p1+i\,q1)\right)+\left(\sqrt{2}\,B+i\,p1-q1\right)\,(i\,p1+q1)}\,\left(-1+r^2\right)\,\text{WgHET}[\delta,0,0,\Delta q,\Delta p]}{8\,\pi},\{0,2,0,2\}\right\},$$

$$\left\{\frac{\sqrt{\frac{3}{2}}\,e^{A\,\left(B+i\,\sqrt{2}\,(p1+i\,q1)\right)+\left(\sqrt{2}\,B+i\,p1-q1\right)\,(i\,p1+q1)}\,r\,\sqrt{1-r^2}\,\text{WgHET}[\delta,0,1,\Delta q,\Delta p]}{4\,\pi},\{0,1,0,2\}\right\},$$

$$\left.\left\{\frac{\sqrt{3}\,e^{A\,B-p1^2-q1^2+\sqrt{2}\,B\,(-i\,p1+q1)-\sqrt{2}\,A\,(i\,p1+q1)}\,r^2\,\text{WgHET}[\delta,0,2,\Delta q,\Delta p]}{8\,\pi},\{0,0,0,2\}\right\}\right\}$$

## $\Pi_{q_2}^{(\theta)}$ on $\left|+_L\right\rangle$

```
In[•]:= ClearAll[WgHET];
(* Define WgHET before defining this and clear the function again *)
WunHETpm0[r_, δ_, Δq_, Δp_, q1_, p1_] = Module[{tmp, res, A, B},
   tmp = ArrayOfWunHETpm0Terms[A, B, r, δ, Δq, Δp, q1, p1];
   res = Total[Table[
      ResolvePD[tmp[[i, 2, 1]], tmp[[i, 2, 2]], A, B, tmp[[i, 1]]]
      , {i, 1, Dimensions[tmp][[1]]}]];
   res];
```

In[∘]:=
```
ClearAll[WgHET];
(* Define WgHET AFTER defining this and clear the function again until ready
 to compute *)PsHETpm0[r_, δ_, Δq_, Δp_] = Module[{tmp, res, A, B, q1, p1},
  tmp = ArrayOfWunHETpm0Terms[A, B, r, δ, Δq, Δp, q1, p1];
  res = Total[Table[
     ResolvePD[tmp[[i, 2, 1]], tmp[[i, 2, 2]], A, B, GaussIntR2[tmp[[i, 1]], q1, p1]]
     , {i, 1, Dimensions[tmp][[1]]}]];
  res];
```

In[∘]:=
```
ClearAll[WgHET];(* Define WgHET AFTER defining this
 and clear the function again until ready to compute *)
fidelityunZPSHETpm0[r_, δ_, Δq_, Δp_] = Module[{tmp1, tmp2, res, q1, p1},
   tmp1 = ArrayOfWunHETpm0Terms[A1, B1, r, δ, Δq, Δp, q1, p1];
   tmp2 = WaltunTHRpmTablepd[A2, B2, 0, 2, 3, r, 1, q1, p1];
   res = Total[Table[ResolvePD2[
        {2 π GaussIntR2[tmp1[[i, 1]] × tmp2[[j, 1]], q1, p1], {tmp1[[i, 2]], tmp2[[j, 2]]}}],
      {i, 1, Dimensions[tmp1][[1]]}, {j, 1, Dimensions[tmp2][[1]]}], 2];
   res];
```

In[13]:=
```
ClearAll[fidelityZPSHETpm0];
fidelityZPSHETpm0[r_, δ_, Δq_, Δp_] :=
       fidelityunZPSHETpm0[r, δ, Δq, Δp]
  ---------------------------------------------------- ;
  PsTHRpm[0, 2, 3, r, 1] × PsHETpm0[r, δ, Δq, Δp]

(*ClearAll[WgHET];
(* Define WgHET AFTER defining this and
 clear the function again until ready to compute *)
purityunHETpm0[r_,δ_,Δq_,Δp_]=Module[{tmp1,tmp2,res,q1,p1},
  tmp1= ArrayOfWunHETpm0Terms[A1,B1,r,δ,Δq,Δp,q1,p1];
  tmp2=ArrayOfWunHETpm0Terms[A2,B2,r,δ,Δq,Δp,q1,p1];
  res =Total[Table[ResolvePD2[
      {2π GaussIntR2[tmp1[[i]][[1]]tmp2[[j]][[1]],q1,p1],{tmp1[[i,2]],tmp2[[j,2]]}}],
     {i,1,Dimensions[tmp1][[1]]},{j,1,Dimensions[tmp2][[1]]}],2];
  res];*)

(*purityHETpm0[r_,δ_,Δq_,Δp_]= purityunHETpm0[r,δ,Δq,Δp] / PsHETpm0[r,δ,Δq,Δp]^2 ;*)
```

### $\Pi_{q_2^{(\theta)}}$ on $|-_L\rangle$

### $\Pi_{q_2^{(\theta)}}$ on $|0_L\rangle$

### $\Pi_{q_2^{(\theta)}}$ on $|1_L\rangle$

## Compiling $W_{un}$, Ps, $\mathcal{F}$ for HET

## [Rslt] $W_{un}$, Ps, & $\mathcal{F}_{un}$ for $\Pi_{HET}|+_L\rangle$

## $W_{un}$

## Ps

## $\mathcal{F}_{un}$

## [Plt] Δ$\mathcal{F}$ & ΔPs

In[319]:=

```
ClearAll[Δℱ, ΔPs];
Δℱpm0[r_, η_, δ_, Δ_] := Re[fidelityZPSHETpm0[r, δ, Δ, Δ] -
    (*from the old code*)FidelityTHR[(*j*)1, 2, 3, r, η]];
ΔPspm0[r_, η_, δ_, Δ_] := Re[PsHETpm0[r, δ, Δ, Δ] - PsTHRpm[0, 2, 3, r, η]];
ClearAll[r, θ, ηH, Δlimit, η, x1, p1,
  x2, p2, plotpoints, Δ, δ, plotROHET, plotRIHET];
r = √.50;
"Dyne efficiency chosen:"
ηH = .85
plotpoints = 60;
Δlimit = 7;
δ = δHET[ηH];
cols =
  Reverse[RGBColor /@ {(*"#053061","#2166ac",*)"#4393c3", "#92c5de", "#d1e5f0",
     "#f7f7f7", "#fddbc7", "#f4a582", "#d6604d"(*,"#b2182b","#67001f"*)}];
"HETERODYNE:"
textb = Text[Style["(b)", {FontSize → Large, FontFamily → "Arial", Black, Bold}],
    {4.5/5 Δlimit, .9}, {0, 0}];
txtb = Graphics[{textb}];
plotROHET = DensityPlot[HeavisideTheta[Δℱpm0[r, η, δ, Δ]],
    {Δ, 0, Δlimit}, {η, 0, 1}, PlotPoints → plotpoints,
    ColorFunction → (Blend[cols, #] &), ColorFunctionScaling → True,
    PlotRange → All, FrameLabel → {"Δ (in units of √ℏ)", "η"}, LabelStyle →
     {FontSize → Large, FontFamily → "Arial", Black}, ImageSize → Medium];
plotRIHET = DensityPlot[HeavisideTheta[ΔPspm0[r, η, δ, Δ]],
    {Δ, 0, Δlimit}, {η, 0, 1}, PlotPoints → plotpoints,
    ColorFunction → (Blend[{RGBColor[0, 0, 0, 0.3], RGBColor[1, 1, 1, 0]}, #] &),
    ColorFunctionScaling → False, PlotRange → All,
    FrameLabel → {"Δ (in units of √ℏ)", "η"}, LabelStyle →
     {FontSize → Large, FontFamily → "Arial", Black}, ImageSize → Medium];
Show[{plotROHET, plotRIHET, txtb}]
ClearAll[r, θ, ηH, Δlimit, plotpoints, Δ, δ, plotROHET, plotRIHET, textb, txtb];
```

Out[324]=

```
Dyne efficiency chosen:
```

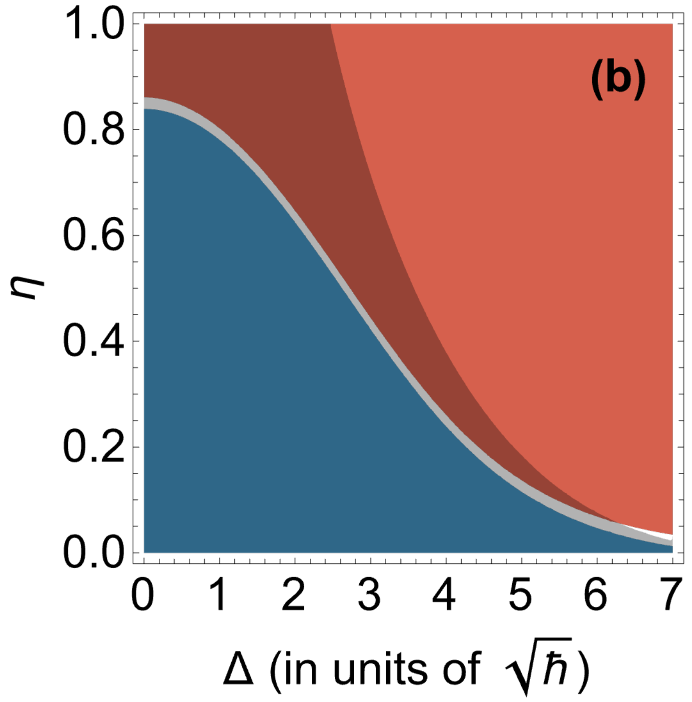
Out[325]=

0.85

Out[330]=

HETERODYNE:

Out[335]=



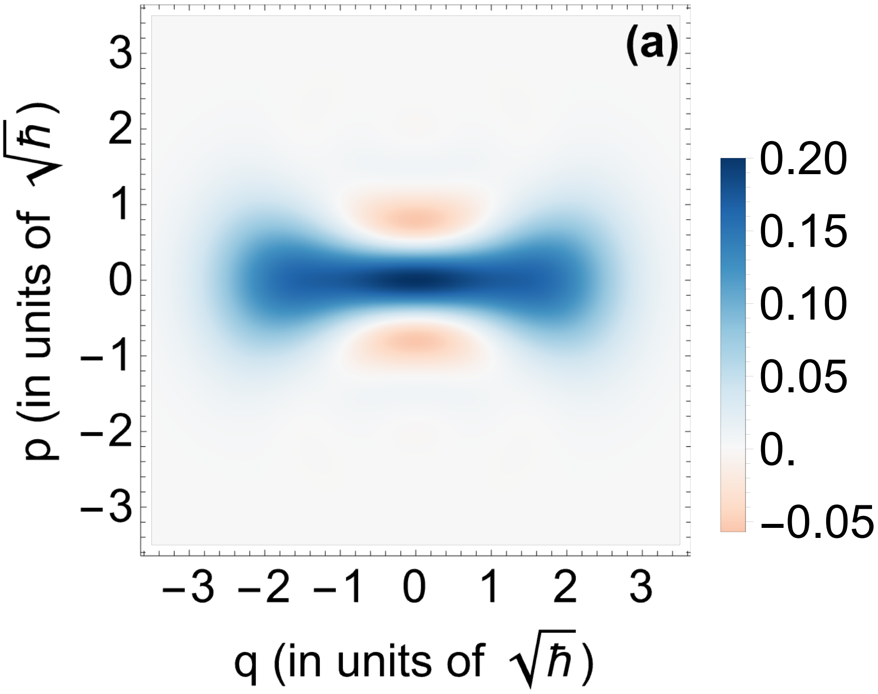

## [Plt] *W*

In[350]:=

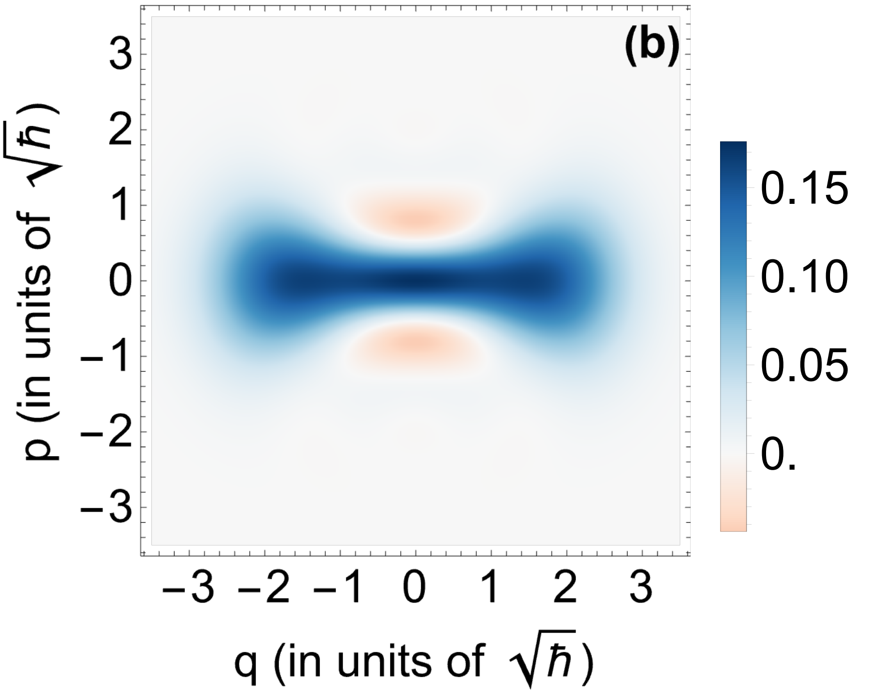
```
(* Options for the density plot *)
cols = RGBColor /@ {"#053061", "#2166ac", "#4393c3", "#92c5de", "#d1e5f0",
    "#f7f7f7", "#fddbc7", "#f4a582", "#d6604d", "#b2182b", "#67001f"};
limits = 3.5;
range = 50;
ClearAll[ηH, η, r, δ, Δq, Δp];
ηH = 0.85;
η  = .75;
r  = √.5 ;
δ  = δHET[ηH];
Δq  = Δp = .75;

ClearAll[mat1, mat2];
(* Construct matrix to plot *)
mat1 = Table[Re[Chop[WunHETpm0[r, δ, Δq, Δp, (q - 1) (limits / range) - limits,
        (p - 1) (limits / range) - limits] / PsHETpm0[r, δ, Δq, Δp]]],
   {p, 1, 2 range + 1}, {q, 1, 2 range + 1}];
(* Normalize the mat *)
mat1 = mat1 / (1 (*Abs[ Total[mat1,2]] (limits/range)^2 *));
```

```
(* Construct matrix to plot *)
mat2 = Table[Re[Chop[WunTHRpm[0, 2, 3, r, η, (q - 1) (limits / range) - limits,
      (p - 1) (limits / range) - limits] / PsTHRpm[0, 2, 3, r, η]]],
   {p, 1, 2 range + 1}, {q, 1, 2 range + 1}];
(* Normalize the mat *)
mat2 = mat2 / 1(*Abs[ Total[mat2,2]] (limits/range)^2*);

(* Plotting *)
texta = Text[Style["(a)", {FontSize → Large, FontFamily → "Arial", Black, Bold}],
   {.9 limits, .9 limits}, {0, 0}];
txta = Graphics[{texta}];
plota = ListDensityPlot[mat1,
   DataRange → {{-limits, limits}, {-limits, limits}}, ColorFunction →
    (Blend[Reverse[cols], (# / Max[{Abs[Max[mat1]], Abs[Min[mat1]]}] + 1) / 2] &),
   ColorFunctionScaling → False, PlotRange → Full,
   FrameLabel → {"q (in units of √ħ)", "p (in units of √ħ)"},
   LabelStyle → {FontSize → Large, FontFamily → "Arial", Black},
   AxesStyle → {Thick, Black}, ImageSize → Medium,
   PlotLegends → Automatic, InterpolationOrder → 1];
Show[{plota, txta}]

textb = Text[Style["(b)", {FontSize → Large, FontFamily → "Arial", Black, Bold}],
   {.9 limits, .9 limits}, {0, 0}];
txtb = Graphics[{textb}];
plotb = ListDensityPlot[mat2,
   DataRange → {{-limits, limits}, {-limits, limits}}, ColorFunction →
    (Blend[Reverse[cols], (# / Max[{Abs[Max[mat2]], Abs[Min[mat2]]}] + 1) / 2] &),
   ColorFunctionScaling → False, PlotRange → Full,
   FrameLabel → {"q (in units of √ħ)", "p (in units of √ħ)"},
   LabelStyle → {FontSize → Large, FontFamily → "Arial", Black},
   AxesStyle → {Thick, Black}, ImageSize → Medium,
   PlotLegends → Automatic, InterpolationOrder → 1];
Show[{plotb, txtb}]
ClearAll[mat1, mat2];
ClearAll[ηH, ηr, r, δ, Δq, Δp];
```

Out[367]=




Out[371]=




## [Rslt] $W_{\text{un}}$, Ps, & $\mathcal{F}_{\text{un}}$ for $\Pi_{\text{HET}} \left| 0_L \right\rangle$