Chapter 1: Quantum measurement

Author: Bruno O. Goes

1) Useful functions

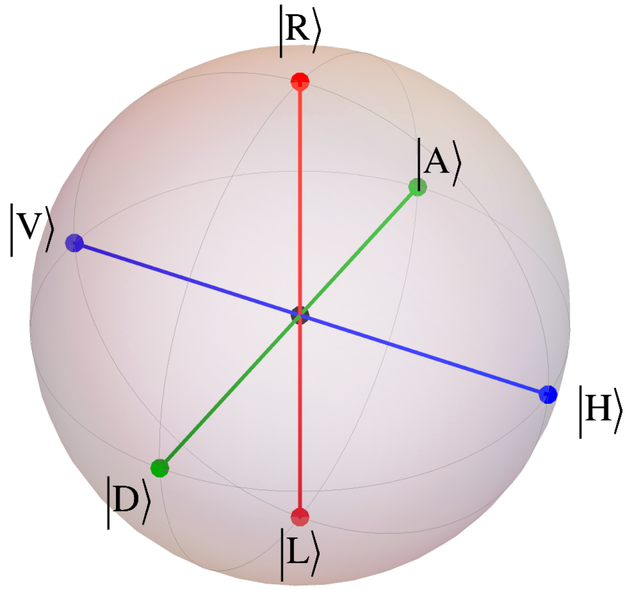
In[62]:=

```
(*Setting the path where the plots are going to be saved.*)
nbdirectory = SetDirectory[NotebookDirectory[]];
PlotsPathChap1 = nbdirectory <> "/PlotsChap1/";

alphabetlabel = {"(a)", "(b)", "(c)", "(d)", "(e)", "(f)", "(g)",
   "(h)", "(i)", "(j)", "(k)", "(l)", "(m)", "(n)", "(o)", "(p)", "(q)",
   "(r)", "(s)", "(t)", "(u)", "(v)", "(w)", "(x)", "(y)", "(z)"};
```

In[19]:=
```
(*Drawing the Poicaré sphere: based on Melt's code*)
PoincareSphere[labels_ : True] :=
 Module[{splineCircle, pointsAndConnection, surroundingCircles, texKet},
  splineCircle[m_List, r_, angles_List : {0, 2 π}] :=
   Module[{seg, ϕ, start, end, pts, w, k}, {start, end} = Mod[N[angles], 2 π];
     If[end ≤ start, end += 2 π];
     seg = Quotient[N[end - start], π/2];
     ϕ = Mod[N[end - start], π/2];
     If[seg == 4, seg = 3;
      ϕ = π/2];
     pts = (r RotationMatrix[start].#1 &) /@ Join[Take[
         {{1, 0}, {1, 1}, {0, 1}, {-1, 1}, {-1, 0}, {-1, -1}, {0, -1}}, 2 seg + 1],
        (RotationMatrix[seg π/2].#1 &) /@ {{1, Tan[ϕ/2]}, {Cos[ϕ], Sin[ϕ]}}];
     If[Length[m] == 2, pts = (m + #1 &) /@ pts, pts = (m + #1 &) /@
        Transpose[Append[Transpose[pts], ConstantArray[0, Length[pts]]]]];
     w = Join[Take[{1, 1/Sqrt[2], 1, 1/Sqrt[2], 1, 1/Sqrt[2], 1}, 2 seg + 1], {Cos[ϕ/2], 1}];
     k = Join[{0, 0, 0}, (Riffle[#1, #1] &) [Range[seg + 1]], {seg + 1}];
     BSplineCurve[pts, SplineDegree → 2, SplineKnots → k, SplineWeights → w]] /;
    Length[m] == 2 || Length[m] == 3;
  pointsAndConnection[points_] :=
   (Sequence @@ {Sequence @@ Point /@ #1, Line[#1]} &) [points];
  surroundingCircles = GeometricTransformation[splineCircle[{0, 0, 0}, 1],
    {{RotationMatrix[0, {1, 0, 0}], {0, 0, 0}}, {RotationMatrix[π/2, {1, 0, 0}],
      {0, 0, 0}}, {RotationMatrix[π/2, {0, 1, 0}], {0, 0, 0}}}];
  texKet[n_] := Text[Style[StringTemplate[
       "\!\(\*TemplateBox[{\"`1`\"},\n\"Ket\"]\)"][
      ToString[n]], FontFamily → "Times", 20]];
  Graphics3D[{White, Opacity[0.3], Sphere[{0, 0, 0}, 1], Opacity[1], Thickness[
     0.004], PointSize[0.02], Red, pointsAndConnection[{{0, 0, 1}, {0, 0, -1}}],
    Blue, pointsAndConnection[{{1, 0, 0}, {-1, 0, 0}}], Darker[Green],
    pointsAndConnection[{{0, 1, 0}, {0, -1, 0}}], Black, Point[{0, 0, 0}],
    If[labels == True, {Text[texKet["R"], {0, 0, 1.2}], Text[texKet["L"], {0, 0,
        -1.2}], Text[texKet["H"], {1.2, 0, 0}], Text[texKet["V"], {-1.2, 0,
        0}], Text[texKet["A"], {0, 1.2, 0}], Text[texKet["D"], {0, -1.2, 0}]}],
    Gray, Thin, surroundingCircles}, Boxed → False, PlotRange →
    ConstantArray[{-1.2, 1.2}, 3], ImageSize → 500, RotationAction → "Clip"]]
```

In[◦]:= PoincareSphere[]

Out[◦]=

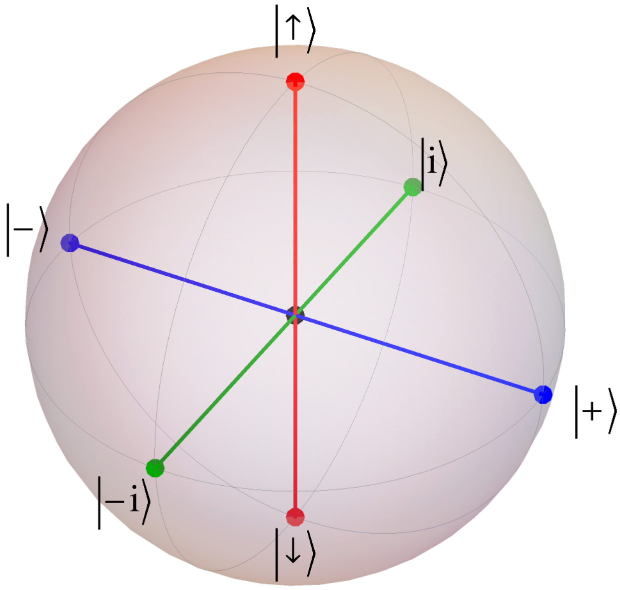


In[20]:=

```
(*Drawing the spin Bloch sphere: based on Melt's code*)
BlochSphereSpin[labels_ : True] :=
 Module[{splineCircle, pointsAndConnection, surroundingCircles, texKet},
  splineCircle[m_List, r_, angles_List : {0, 2 π}] :=
   Module[{seg, ϕ, start, end, pts, w, k}, {start, end} = Mod[N[angles], 2 π];
     If[end ≤ start, end += 2 π];
     seg = Quotient[N[end - start], π/2];
     ϕ = Mod[N[end - start], π/2];
     If[seg == 4, seg = 3;
      ϕ = π/2];
     pts = (r RotationMatrix[start].#1 &) /@ Join[Take[
         {{1, 0}, {1, 1}, {0, 1}, {-1, 1}, {-1, 0}, {-1, -1}, {0, -1}}, 2 seg + 1],
        (RotationMatrix[seg π/2].#1 &) /@ {{1, Tan[ϕ/2]}, {Cos[ϕ], Sin[ϕ]}}];
     If[Length[m] == 2, pts = (m + #1 &) /@ pts, pts = (m + #1 &) /@
        Transpose[Append[Transpose[pts], ConstantArray[0, Length[pts]]]]];
     w = Join[Take[{1, 1/Sqrt[2], 1, 1/Sqrt[2], 1, 1/Sqrt[2], 1}, 2 seg + 1], {Cos[ϕ/2], 1}];
     k = Join[{0, 0, 0}, (Riffle[#1, #1] &) [Range[seg + 1]], {seg + 1}];
     BSplineCurve[pts, SplineDegree → 2, SplineKnots → k, SplineWeights → w]] /;
    Length[m] == 2 || Length[m] == 3;
  pointsAndConnection[points_] :=
   (Sequence @@ {Sequence @@ Point /@ #1, Line[#1]} &) [points];
  surroundingCircles = GeometricTransformation[splineCircle[{0, 0, 0}, 1],
    {{RotationMatrix[0, {1, 0, 0}], {0, 0, 0}}, {RotationMatrix[π/2, {1, 0, 0}],
      {0, 0, 0}}, {RotationMatrix[π/2, {0, 1, 0}], {0, 0, 0}}}];
  texKet[n_] := Text[Style[StringTemplate[
       "\!\(\*TemplateBox[{\"`1`\"},\n\"Ket\"]\)"][
      ToString[n]], FontFamily → "Times", 20]];
  Graphics3D[{White, Opacity[0.3], Sphere[{0, 0, 0}, 1], Opacity[1], Thickness[
     0.004], PointSize[0.02], Red, pointsAndConnection[{{0, 0, 1}, {0, 0, -1}}],
    Blue, pointsAndConnection[{{1, 0, 0}, {-1, 0, 0}}], Darker[Green],
    pointsAndConnection[{{0, 1, 0}, {0, -1, 0}}], Black, Point[{0, 0, 0}],
    If[labels == True, {Text[texKet["↑"], {0, 0, 1.2}], Text[texKet["↓"], {0, 0,
         -1.2}], Text[texKet["+"], {1.2, 0, 0}], Text[texKet["-"], {-1.2, 0,
         0}], Text[texKet["i"], {0, 1.2, 0}], Text[texKet["-i"], {0, -1.2, 0}]}],
    Gray, Thin, surroundingCircles}, Boxed → False, PlotRange →
    ConstantArray[{-1.2, 1.2}, 3], ImageSize → 500, RotationAction → "Clip"]]
```

In[ ]:=
```
BlochSphereSpin[]
```

Out[ ]=



2)Loading the necessary library

This notebook uses Melt! library developed by Gabriel Landi (available at https://melt.super.site/). It has useful built-in quantum information functions. In order for this notebook to work properly it is necessary to uncomment and run the following cell:

```
(*Get["http://www.fmt.if.usp.br/~gtlandi/download/melt.m"]
 (*Loads Melt! on Mathematica*)
 LoadPauliMatrices[](*Loads Pauli Matrices σj*)*)
```

This section contains the plots of important quantities based on the expressions presented in the chapter 1 of the thesis.

1.von Neumann measurement model

In this section I use the calculations of the chapter 1 of the thesis for a spin 1/2 and a spin 3/2 particle.

Loading the necessary spin matrices:

In[21]:=
```
LoadPauliMatrices[]
LoadArbitrarySpinMatrices[3 / 2]
```

Out[21]= Matrices loaded: σ0 (=1), σx, σy, σz, σp, σm, GMAT[θ,ϕ]

Out[22]= Matrices loaded: S0 (=1), Sx, Sy, Sz, Sp, Sm

Computing the eigenvalues and eigenvectors:

In[23]:=
```
{eigenvaluesσzlist, eigenvectorsσzlist} = Eigensystem[σz];
{eigenvaluesSzlist, eigenvectorsSzlist} = Eigensystem[Sz];
```

Defining the initial and final states of the target system, the only variable is the product $g_0\, t$:

In[25]:=
```
ψ01half = Sum[(-1)^i eigenvectorsσzlist[[i]], {i, 1, Length@eigenvectorsσzlist}];
ψ03half = Sum[(-1)^i eigenvectorsSzlist[[i]], {i, 1, Length@eigenvectorsSzlist}];
ψ01half = ψ01half / Sqrt[ψ01half.ψ01half] // mf;
ψ03half = ψ03half / Sqrt[ψ03half.ψ03half] // mf;
```

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\begin{pmatrix} 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \end{pmatrix}$$

Defining the initial and final density matrices of the target system, the only variable is the product $g_0\, t$:

In[29]:=
```
ρ01half = out[ψ01half, ψ01half*] // mf;
ρ03half = out[ψ03half, ψ03half*] // mf;
```

$$\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

$$\begin{pmatrix} 0.25 & 0.25 & -0.25 & -0.25 \\ 0.25 & 0.25 & -0.25 & -0.25 \\ -0.25 & -0.25 & 0.25 & 0.25 \\ -0.25 & -0.25 & 0.25 & 0.25 \end{pmatrix}$$

In[31]:=
```
Tr[ρ01half]
Tr[ρ03half]
```

Out[31]= 1

Out[32]= 1.

Computing the density matrices at time t:

In[33]:=
$$\rho t1half = \frac{1}{2} \text{Sum}\Big[\text{Exp}\Big[- \frac{g0t^2}{2} (\text{eigenvalues}\sigma\text{zlist}[[k]] - \text{eigenvalues}\sigma\text{zlist}[[q]])^2\Big]$$
```
out[eigenvectorsσzlist[[k]], eigenvectorsσzlist[[q]]],
{k, 1, Length@eigenvaluesσzlist}, {q, 1, Length@eigenvaluesσzlist}] // mf;
```
$$\rho t3half = \frac{1}{4} \text{Sum}\Big[\text{Exp}\Big[- \frac{g0t^2}{2} (\text{eigenvaluesSzlist}[[k]] - \text{eigenvaluesSzlist}[[q]])^2\Big]$$
```
out[eigenvectorsSzlist[[k]], eigenvectorsSzlist[[q]]],
{k, 1, Length@eigenvaluesSzlist}, {q, 1, Length@eigenvaluesSzlist}] // mf;
```

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} e^{-2\, g0t^2} \\ \frac{1}{2} e^{-2\, g0t^2} & \frac{1}{2} \end{pmatrix}$$

$$\begin{pmatrix} 0.25 & 0.25\, e^{-0.5\, g0t^2} & 0.25\, e^{-2.\, g0t^2} & 0.25\, e^{-4.5\, g0t^2} \\ 0.25\, e^{-0.5\, g0t^2} & 0.25 & 0.25\, e^{-0.5\, g0t^2} & 0.25\, e^{-2.\, g0t^2} \\ 0.25\, e^{-2.\, g0t^2} & 0.25\, e^{-0.5\, g0t^2} & 0.25 & 0.25\, e^{-0.5\, g0t^2} \\ 0.25\, e^{-4.5\, g0t^2} & 0.25\, e^{-2.\, g0t^2} & 0.25\, e^{-0.5\, g0t^2} & 0.25 \end{pmatrix}$$

In[35]:=
```
Tr@ρt1half
Tr@ρt3half
```
Out[35]= 1

Out[36]= 1.

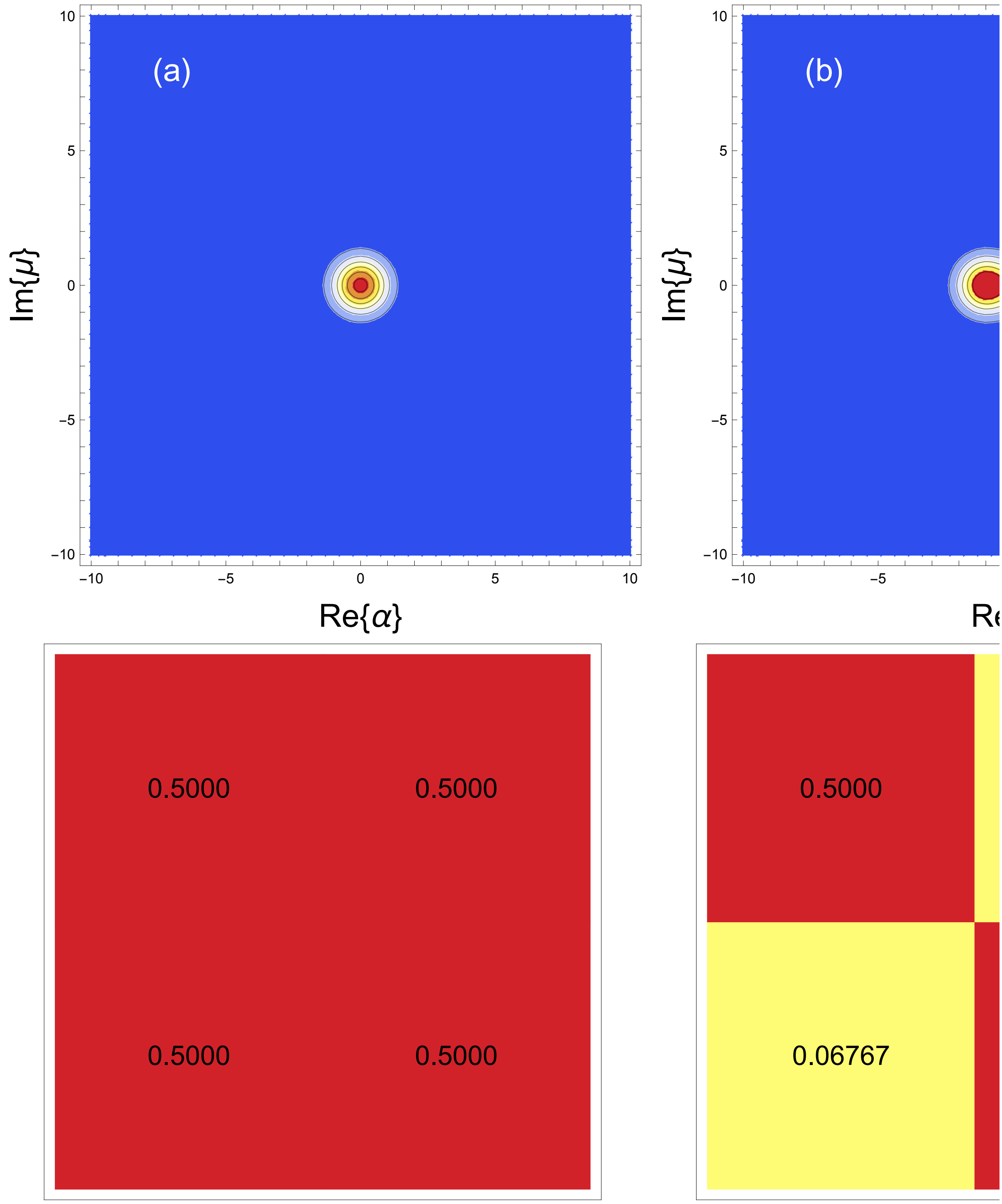
Computing the Q-function:

In[48]:=
$$\text{Q1half} = \text{Sum}\Big[\frac{1}{\pi} \left(\text{Exp}\big[-\text{Abs}[(\text{re}\mu + i\, \text{im}\mu) - \text{g0t bk}]^2\big]\right), \{\text{bk}, \text{eigenvalues}\sigma\text{zlist}\}\Big];$$
$$\text{Q3half} = \text{Sum}\Big[\frac{1}{4\pi} \left(\text{Exp}\big[-\text{Abs}[(\text{re}\mu + i\, \text{im}\mu) - \text{g0t bk}]^2\big]\right), \{\text{bk}, \text{eigenvaluesSzlist}\}\Big];$$

Plotting the target system state and the Q-function associated to it:

In[99]:=
```
vNmodelExamplePlot1half = Grid[
   {
    (*First row - Contour plot of the Q-function*)
    Table[Quiet@ContourPlot[
       Quiet@Q1half /. g0t → c /. reμ → x /. imμ → y, {x, -10, 10}, {y, -10, 10},
       Epilog → Inset[Style[alphabetlabel[[c + 1]], White, FontSize → 20], {-7, 8}],
       PlotPoints → 40, PlotRange → All, ImageSize → 400,
       ColorFunction → "TemperatureMap", FrameStyle → Black,
       FrameLabel → {Style["Re{α}", Black, FontSize → 20],
         Style["Im{μ}", Black, FontSize → 20]},
       AxesLabel → {{Style["Re{μ}", Black, FontSize → 20], None}, Style["Im{α}",
          Black, FontSize → 20]}, TicksStyle → Directive[Black, FontSize → 20]
       (*Set ticks style to black and fontsize 20*)], {c, {0, 1, 2, 3}}]
    ,

    (*Second row - Hinton diagram of the density matrix*)
    Table[MatrixPlot[ρt1half /. g0t → c,
      FrameTicks → None,
      ColorFunction → "TemperatureMap",
      ImageSize → 350,
      Epilog → Table[Text[Style[Chop@N[ρt1half[[i, j]] /. g0t → c, 4], 17],
          {j - 0.5, Length[ρt1half] - i + 0.5}],
         {i, Length[ρt1half]}, {j, Length[ρt1half[[i]]]}]
     ], {c, {0, 1, 2, 5}}]
   }

  ]
Export[PlotsPathChap1 <> "vNmodelExample1half.pdf", vNmodelExamplePlot1half];
```

Out[99]=


(a)
(b)
Im{μ}
Im{μ}
Re{α}
Re
10
5
0
−5
−10
0.5000
0.5000
0.5000
0.5000
0.5000
0.06767

In[67]:=
$$\text{Qplusonehalf} = \frac{1}{\pi}\left(\text{Exp}\left[-\text{Abs}\left[(\text{re}\mu + i\,\text{im}\mu) - \frac{\text{g0t}}{2}\right]^2\right]\right) \;//\; \text{cf}$$
$$\text{Qminusonehalf} = \frac{1}{\pi}\left(\text{Exp}\left[-\text{Abs}\left[(\text{re}\mu + i\,\text{im}\mu) + \frac{\text{g0t}}{2}\right]^2\right]\right) \;//\; \text{cf}$$

Out[67]=
$$\frac{e^{-\text{Abs}\left[-\frac{\text{g0t}}{2}+i\,\text{im}\mu+\text{re}\mu\right]^2}}{\pi}$$

Out[68]=
$$\frac{e^{-\text{Abs}\left[\frac{\text{g0t}}{2}+i\,\text{im}\mu+\text{re}\mu\right]^2}}{\pi}$$

In[69]:=
```
Integrate[Integrate[Qplusonehalf, {reμ, -20, 20}], {imμ, -20, 20}]
```

Out[69]=
$$\frac{1}{4}\left(\text{Erf}\left[20-\frac{\text{Im}[\text{g0t}]}{2}\right]+\text{Erf}\left[20+\frac{\text{Im}[\text{g0t}]}{2}\right]\right)\left(\text{Erf}\left[20-\frac{\text{Re}[\text{g0t}]}{2}\right]+\text{Erf}\left[20+\frac{\text{Re}[\text{g0t}]}{2}\right]\right)$$

In[70]:=
$$\frac{1}{4}\left(\text{Erf}\left[20-\frac{\text{Im}[\text{g0t}]}{2}\right]+\text{Erf}\left[\frac{1}{2}\times(40+\text{Im}[\text{g0t}])\right]\right)\left(\text{Erf}\left[20-\frac{\text{Re}[\text{g0t}]}{2}\right]+\text{Erf}\left[20+\frac{\text{Re}[\text{g0t}]}{2}\right]\right) \;//\; \text{cf}$$

Out[70]=
$$\frac{1}{2}\,\text{Erf}[20]\left(\text{Erf}\left[20-\frac{\text{g0t}}{2}\right]+\text{Erf}\left[20+\frac{\text{g0t}}{2}\right]\right)$$

In[71]:=
$$\text{argOfintegral} = \sqrt{\text{Qplusonehalf Qminusonehalf}} \;//\; \text{cf}$$

Out[71]=
$$\frac{e^{-\frac{\text{g0t}^2}{4}-\text{im}\mu^2-\text{re}\mu^2}}{\pi}$$

In[72]:=
```
cBhatToyModel =
  Integrate[Integrate[argOfintegral, {reμ, -20, 20}], {imμ, -20, 20}]
```

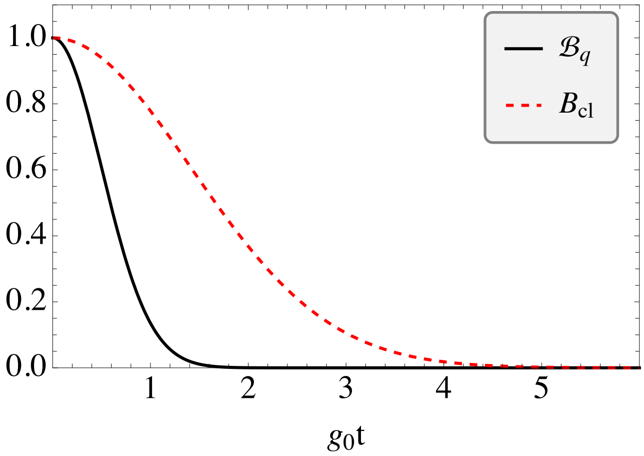
Out[72]=
$$e^{-\frac{\text{g0t}^2}{4}}$$

In[73]:=
```
qℬhatToyModel = Abs[Tr[σx.ρt1half]] // cf
```

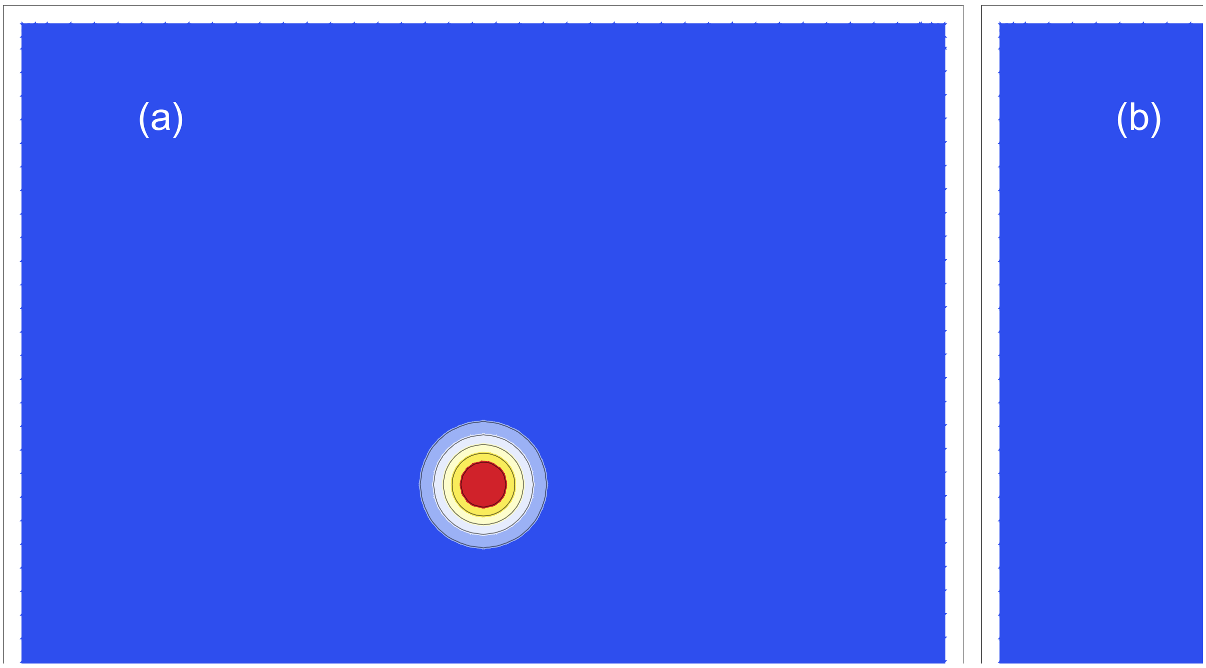
Out[73]=
$$e^{-2\,\text{g0t}^2}$$

In[74]:=
```
ToyModelcandqBhat = Plot[{qℬhatToyModel, cBhatToyModel}, {g0t, 0, 6},
  PlotStyle → {{Black}, {Dashed, Red}},
  PlotRange → {All, {0, 1.1}},
  PlotLegends → legend[{"ℬq", "Bcl"}, {0.85, 0.8}],
  FrameLabel → {"g0t", None}]
Export[PlotsPathChap1 <> "ToyModelcandqBhat1half.pdf", ToyModelcandqBhat];
```

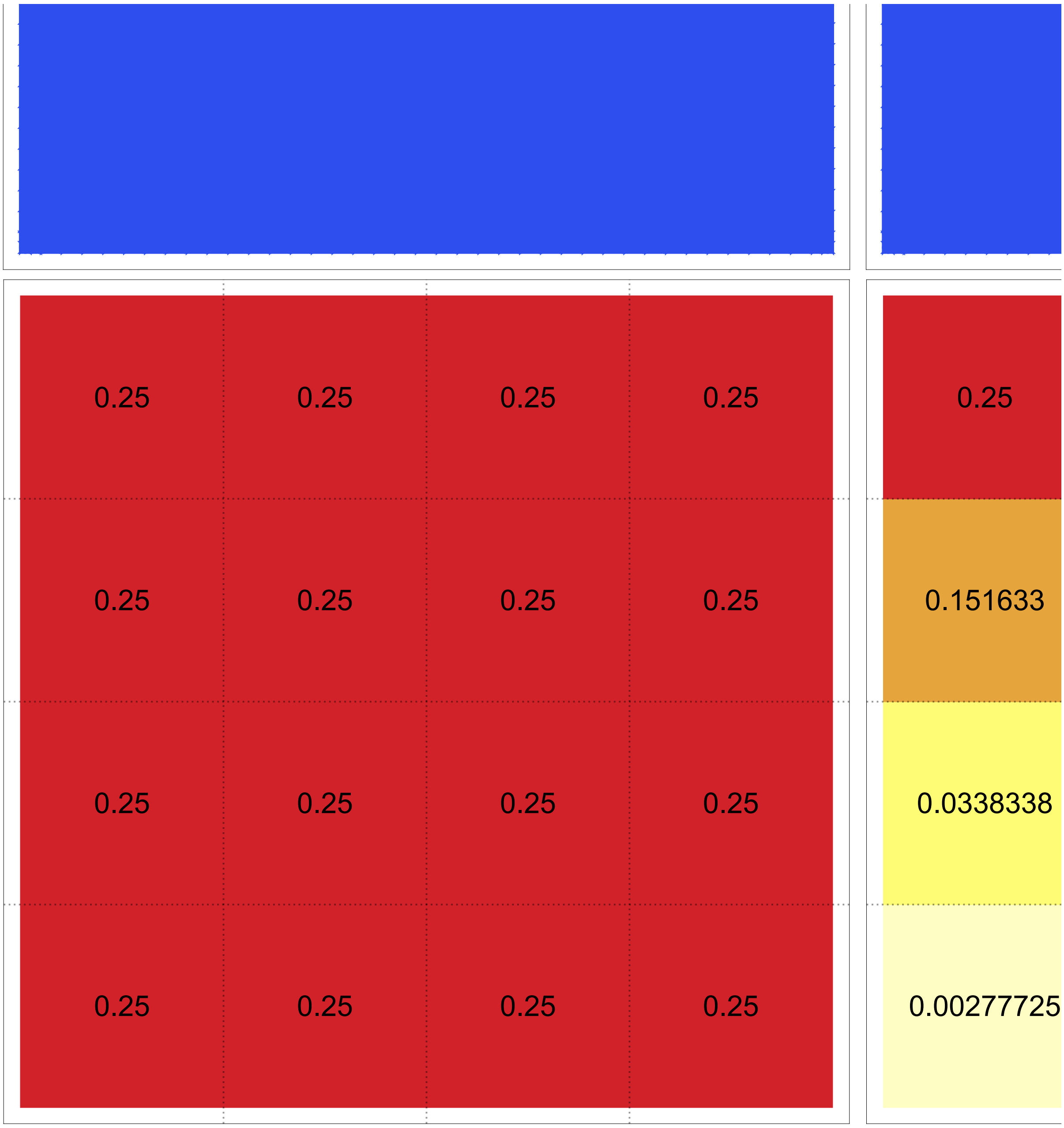
Out[74]=



In[82]:=
```
vNmodelExamplePlot3half = Grid[
  {
   (*First row - Contour plot of the Q-function*)
   Table[
    Quiet@ContourPlot[
      Quiet@Q3half /. g0t → c /. reμ → x /. imμ → y, {x, -10, 10}, {y, -10, 10},
      Epilog → Inset[Style[alphabetlabel[[c + 1]], White, FontSize → 20], {-7, 8}],
      PlotPoints → 40,
      ImageSize → 500,
      PlotRange → All,
      ColorFunction → "TemperatureMap",
      FrameTicks → None,
      AxesLabel → {"Re{α}", "Im{α}"}], {c, {0, 1, 2, 3, 4}}],

   (*Second row - Hinton diagram of the density matrix*)
   Table[MatrixPlot[ρt3half /. g0t → c,
     ColorFunction → "TemperatureMap",
     PlotTheme → "Business",
     ImageSize → 500,
     FrameTicks → None,
     Epilog → Table[Text[Style[Chop@N[ρt3half[[i, j]] /. g0t → c, 4], 17],
        {j - 0.5, Length[ρt3half] - i + 0.5}],
       {i, Length[ρt3half]}, {j, Length[ρt3half[[i]]]}],
     FrameTicks → {{1, MX@" "}, {2, MX@" "}, {3, MX@" "}, {4, MX@" "}}],
    {c, {0, 1, 2, 3, 4}}]

  }

 ]
Export[PlotsPathChap1 <> "vNmodelExample3half.pdf", vNmodelExamplePlot3half];
```



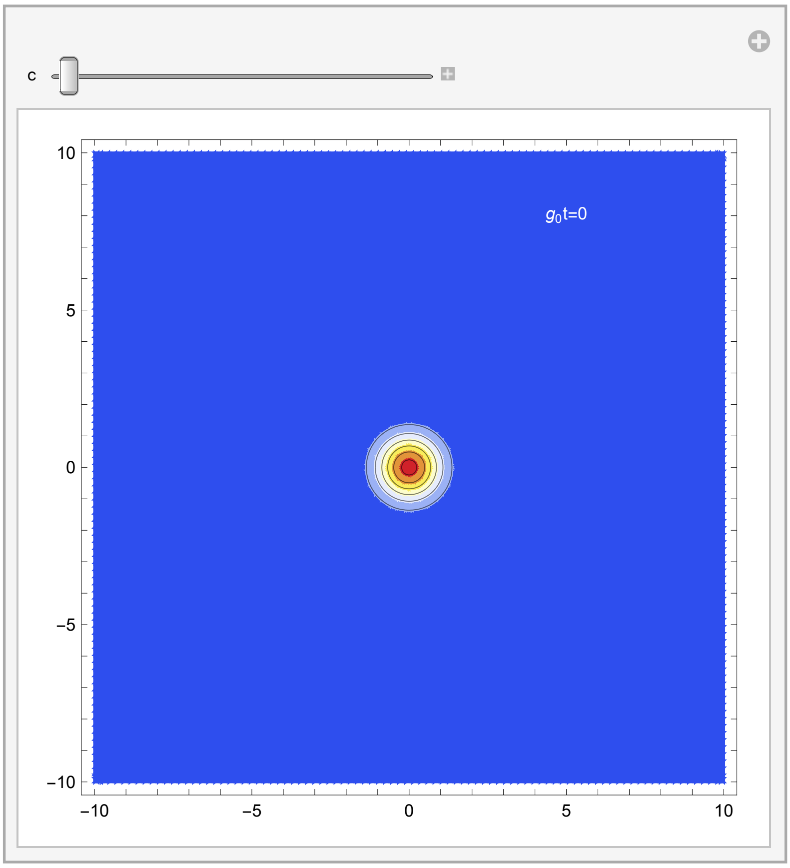
Out[82]=



Playing with Manipulate:

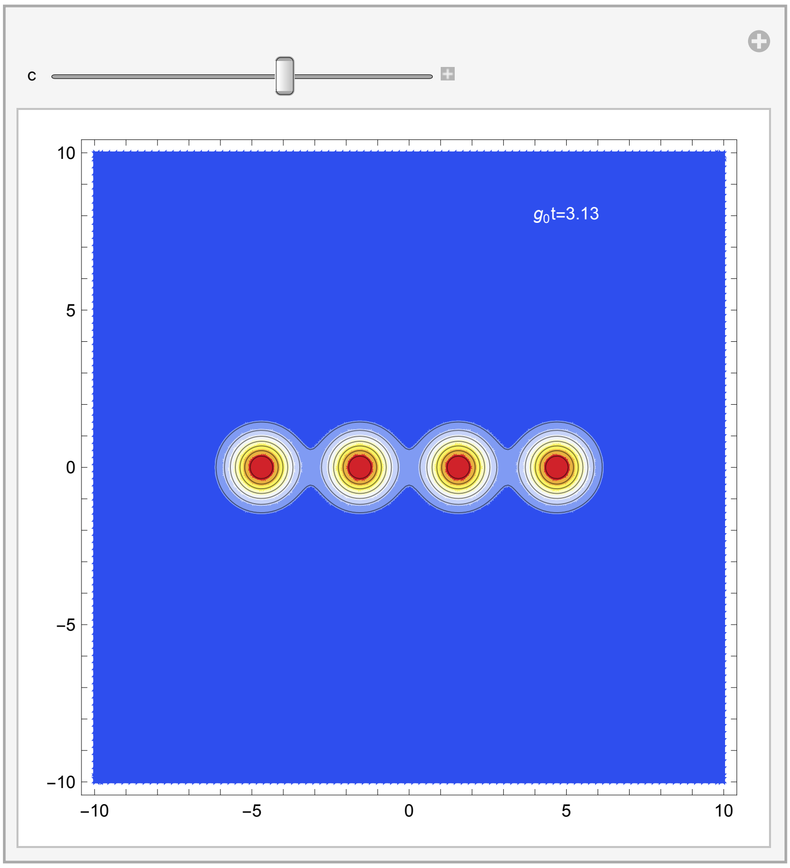
In[101]:=
```
Manipulate[
 ContourPlot[Quiet@Q1half /. g0t → c /. reμ → x /. imμ → y,
  {x, -10, 10}, {y, -10, 10}, PlotPoints → 90, PlotRange → All,
  ColorFunction → "TemperatureMap", AxesLabel → {"Re{α}", "Im{α}"},
  Epilog → Inset[Style["g₀t=" <> ToString[c], White], {5, 8}]], {c, 0, 5, 0.01}]
```

Out[101]=



In[102]:=
```
Manipulate[
 ContourPlot[Quiet@Q3half /. g0t → c /. reμ → x /. imμ → y,
  {x, -10, 10}, {y, -10, 10}, PlotPoints → 90, PlotRange → All,
  ColorFunction → "TemperatureMap", AxesLabel → {"Re{α}", "Im{α}"},
  Epilog → Inset[Style["g₀t=" <> ToString[c], White], {5, 8}]], {c, 0, 5, 0.01}]
```

Out[102]=



Making cool gifs:

In[110]:=
```
PrettyTiming[
 AnimQfunc = Table[Quiet@ContourPlot[Quiet@Q1half /. g0t → c /. reμ → x /. imμ → y,
     {x, -10, 10}, {y, -10, 10}, PlotPoints → 40, PlotRange → All, ColorFunction →
      "TemperatureMap", AxesLabel → {"Re{α}", "Im{α}"}, Epilog → Inset[
       Style["g₀t=" <> ToString[c], FontSize → 30], {5, 8}]], {c, 0, 5, 0.1}];
 Animρt = Table[
   MatrixPlot[ρt1half /. g0t → c,
    FrameTicks → None,
    ColorFunction → "TemperatureMap",
    ImageSize → 350,
    Epilog → Table[Text[Style[Chop@N[ρt1half⟦i, j⟧ /. g0t → c, 4], 17],
       {j - 0.5, Length[ρt1half] - i + 0.5}],
      {i, Length[ρt1half]}, {j, Length[ρt1half⟦i⟧]}]
   ],
   {c, 0, 5, 0.1}];]
```

0h : 0m : 9s

In[111]:=
```
Export[PlotsPathChap1 <> "GifQfunction1half.gif", AnimQfunc];
Export[PlotsPathChap1 <> "GifDensityMatrix1half.gif", Animρt];
```

In[116]:=
```
PrettyTiming[
 AnimQfunc = Table[Quiet@ContourPlot[Quiet@Q3half /. g0t → c /. reμ → x /. imμ → y,
     {x, -10, 10}, {y, -10, 10}, PlotPoints → 40, PlotRange → All,
     ColorFunction → "TemperatureMap", AxesLabel → {"Re{α}", "Im{α}"},
     Epilog → Inset["g₀t=" <> ToString[c], {5, 8}]], {c, 0, 5, 0.1}];
 Animρt = Table[
   MatrixPlot[ρt3half /. g0t → c,
    ColorFunction → "TemperatureMap",
    PlotTheme → "Business",
    ImageSize → 500,
    FrameTicks → None,
    Epilog → Table[Text[Style[Chop@N[ρt3half⟦i, j⟧ /. g0t → c, 4], 17],
       {j - 0.5, Length[ρt3half] - i + 0.5}],
      {i, Length[ρt3half]}, {j, Length[ρt3half⟦i⟧]}],
    FrameTicks → {{1, MX@" "}, {2, MX@" "}, {3, MX@" "}, {4, MX@" "}}],
   {c, 0, 5, 0.1}];]
```

0h : 0m : 18s

In[117]:=
```
Export[PlotsPathChap1 <> "GifQfunction3half.gif", AnimQfunc];
Export[PlotsPathChap1 <> "GifDensityMatrix3half.gif", Animρt];
```