Chapter 4 pt. 2: Photon-photon gate and error analysis

Author: Bruno Ortega Goes

Preamble

In[313]:=

```
PlotsPath = "/Users/brunogoes/Dropbox/00--Thesis/Figures/Chap5/";
overleafPath =
  "/Users/brunogoes/Dropbox/Aplicativos/Overleaf/00Bruno-Goes-ThesisDraft/
    Figures/Chap5/";
overleafPath = "/Users/brunogoes/Dropbox/Aplicativos/";

nbdirectory = SetDirectory[NotebookDirectory[]];
plotsPath = nbdirectory <> "/PlotsChap4/";

Clear[savePlot];
savePlot[nameAndExtension_, plot_, path_ : plotsPath] :=
 If[Length@path == 0, Export[path <> nameAndExtension, plot], Do[
   Export[PlotsPath <> nameAndExtension, plot], {PlotsPath, path}]]

fontsize = 24;

(*To easily insert a label*)
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)"};

AlphabetLabelling[x_, xpos_ : 0.1, ypos_ : 0.9] :=
  Text["(" <> ToString@x <> ")", Scaled[{xpos, ypos}]];
```

We define the ordered Pauli basis:

In[310]:=
```
𝒫 = Table[kron[i, j], {i, {σ0, σx, σy, σz}}, {j, {σ0, σx, σy, σz}}] // mf;
```

$$\begin{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & -i \\ i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & -i & 0 \\ 0 & i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & -i & 0 \\ 0 & 0 & 0 & i \\ i & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 0 & -i & 0 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{pmatrix}$$

In[311]:=
```
Clear[Fave];
```

$$\text{Fave[Ureal\_, Uideal\_]} := \frac{\text{Tr[Ureal}^{\dagger}\text{.Ureal]} + \text{Abs[Tr[Uideal}^{\dagger}\text{.Ureal]]}^2}{20}$$

Pulse shape:Decreasing pulse

Through this section I apply the formalism assuming a decreasing exponential pulse.

Input and output pulse shapes

In[323]:=
```
Intensity[funcoft_] := funcoft funcoft* // cf
```

In[324]:=
```
Clear[ξdec, γ, Γ, ω0];
```

$$\text{ξdec[t\_]} := \sqrt{\Gamma}\ \text{Exp}\left[-\left(\frac{\Gamma}{2}\right) t\right];$$

```
InputIntensityDec = Intensity[ξdec[t]];

Assuming[Γ > 0, Integrate[ξdec[t] × ξdec[t]* // cf, {t, 0, ∞}]]
```

Out[327]= 1

In[328]:=
```
Clear[ξdectilde];
ξdectilde[t_] := e^(-γt/2) Integrate[e^((γ/2) s) ξdec[s], {s, 0, t}];
ξdectilde[t]
```

Out[330]=
$$\frac{2\, e^{-\frac{t\gamma}{2}} \left(-1 + e^{\frac{1}{2} t (\gamma-\Gamma)}\right) \sqrt{\Gamma}}{\gamma - \Gamma}$$

In[331]:=
```
(*Computing the output shape*)
ϒdec[t_] := γ ξdectilde[t] - ξdec[t]
OutputIntensityDec = Intensity[ϒdec[t]];
Assuming[{Γ > 0 && γ > 0}, Integrate[ϒdec[t] × ϒdec[t]* // cf, {t, 0, ∞}]]
```

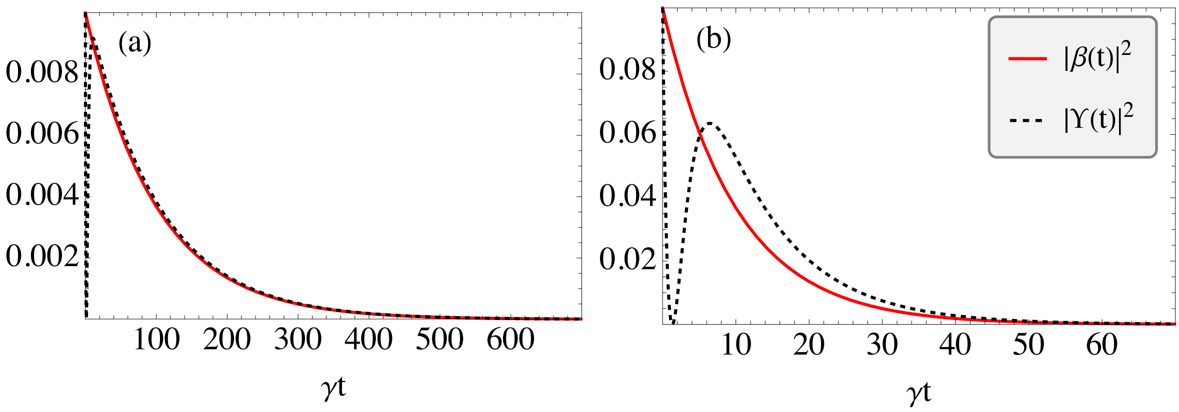
Out[333]= 1

In[334]:=
```
ϒdec[t] // cf
```

Out[334]=
$$\frac{e^{-\frac{1}{2} t (\gamma+\Gamma)} \sqrt{\Gamma} \left(2\, e^{\frac{t\Gamma}{2}} \gamma - e^{\frac{t\gamma}{2}} (\gamma + \Gamma)\right)}{-\gamma + \Gamma}$$

In[344]:=
```
γ = 1;

IntensityCurvesLines = {{Red, Automatic}, {Black, Dashing[{.01}]}};

Grid[
 {
  {fig1a = Plot[
     {InputIntensityDec /. Γ → 10^-2, OutputIntensityDec /. Γ → 10^-2},
     {t, 0, 7 Γ^-1} /. Γ → 10^-2,
     Frame → True,
     PlotStyle → IntensityCurvesLines,
     PlotRange → All,
     FrameStyle → Black,
     ImageSize → 300,
     FrameLabel → {"γt", None},
     Epilog → AlphabetLabelling["a"]],

   fig1b = Plot[{InputIntensityDec /. Γ → 10^-1,
      OutputIntensityDec /. Γ → 10^-1}, {t, 0, 7 Γ^-1} /. Γ → 10^-1,
     Frame → True,
     PlotStyle → IntensityCurvesLines,
     PlotRange → All,
     FrameStyle → Black,
     ImageSize → 300,
     FrameLabel → {"γt", None},
     PlotLegends → legend[{"|β(t)|^2", "|Υ(t)|^2"}, {0.8, 0.75}],
     Epilog → AlphabetLabelling["b"]]}

  (*Plot[

    {InputIntensityDec/.Γ→1, OutputIntensityDec/.Γ→1},{t,0,7 Γ^-1}/.Γ→1,
    Frame→True,
    PlotStyle→IntensityCurvesLines,
    PlotRange→All,
    FrameStyle→Black,
    ImageSize→300,
    FrameLabel→{"γt",None},
    PlotLegends→legend[{"|ξ(t)|^2","|υ(t)|^2"},{0.8,0.75}],
    Epilog→AlphabetLabelling["c"]]}
  *)
 }]
savePlot["Chap5DecreasingPulseShapeDiffeGammas.pdf", %];
Export[overleafPath <> "Chap5DecreasingPulseShapeDiffeGammas.pdf", %];
```

Out[346]=



Non-monochromaticity parameter

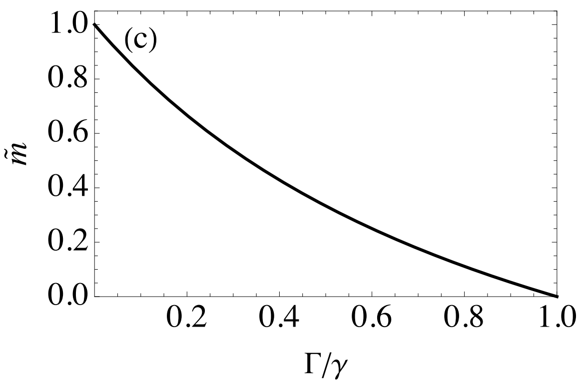
In[351]:=
```
m[InputOfT_, OutputOfT_] := Module[{dummy},
  dummy = InputOfT* OutputOfT // cf;
  Integrate[dummy, {t, 0, ∞}] // cf]
```

In[352]:=
```
Clear[γ]
mdec = m[ξdec[t], ϒdec[t]]
```

Out[353]= $-1+\frac{2\gamma}{\gamma+\Gamma}$

In[354]:=
```
fig1c = Plot[mdec /. γ → 1, {Γ, 0, 2},
   Frame → True,
   PlotStyle → {Black},
   PlotRange → {All, {0, 1.05}},
   FrameStyle → Black,
   ImageSize → 300,
   FrameLabel → {"Γ/γ", "m̃"},
   Epilog → AlphabetLabelling["c"]
  ]
Export[overleafPath <> "Chap5DecreasingNonMonochromaticity.pdf", %];
```

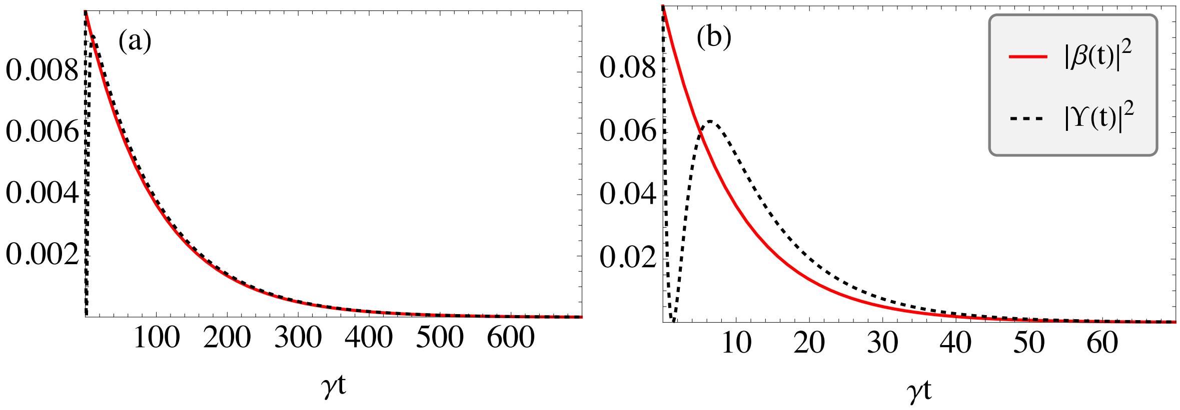
Out[354]=



Based on forthcoming analysis, the gate is implemented when m≥0.4. From this plot we have that the minimum value of Γ=0.8γ.

In[356]:=
```
Grid[{{fig1a, fig1b, fig1c}}]
savePlot["Chap5Fig1.pdf", %];
Export[overleafPath <> "Chap5Fig1.pdf", %%];
```

Out[356]=



In[359]:= Solve$\left[-1+\frac{2\gamma}{\gamma+\Gamma}==0.4, \{\Gamma\}\right]$

Solve: Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

Out[359]= $\{\{\Gamma \to 0.428571\,\gamma\}\}$

Error analysis

Analysis for the gate conditioned on spin ↑

In[360]:= GuprealOriginal = $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{m^2+2m-1}{2} \end{pmatrix}$ // cf;

GuprealOriginal /. m → 1 // mf;

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

The non-monochromaticity introduces a phase error to the desired gate. We parametrize this error by Exp[*i* ϵup] where ϵup is the gate angle error.

The real rotation, when the scattered field is not monochromatic is given by:

In[362]:= sol1 = Flatten@Solve$\left[\left\{\frac{m^2+2m-1}{2} == e^{i\,\epsilon up}\right\}, \epsilon up\right]$ // cf

Out[362]= $\left\{\epsilon up \to 2\pi c_1 - i \,\text{Log}\left[\frac{1}{2}\times(-1+m(2+m))\right] \text{ if } c_1 \in \mathbb{Z}\right\}$

Since we want $e^{i\,\epsilon up}$=1, c_1=1.

In[363]:=
```
gateUpangleerror = ⅇ^(ⅈ ϵup) /. ϵup → 2 π - ⅈ Log[1/2 × (-1 + m (2 + m))]
```

Out[363]= $e^{i\left(2\pi - i \,\mathrm{Log}\left[\frac{1}{2}\times(-1+m\,(2+m))\right]\right)}$

If m=1

In[364]:=
```
gateUpangleerror /. m → 1
```

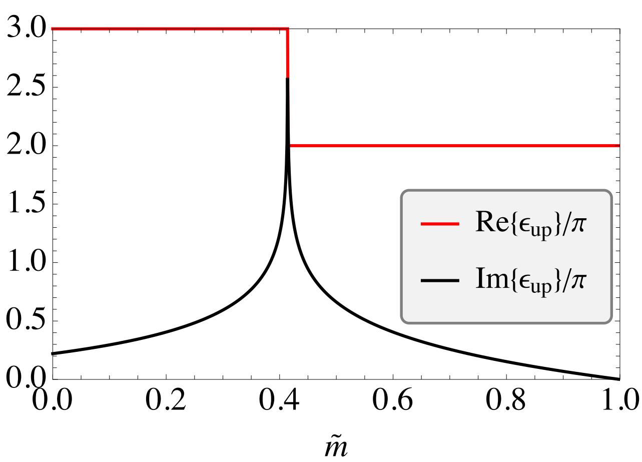
Out[364]= 1

In[365]:=
```
realGateAngle = 2 π - ⅈ Log[1/2 × (-1 + m (2 + m))];
```

In[366]:=
```
imaginaryparError = Table[{m, Chop[Im[realGateAngle]]/π}, {m, 0, 1, 0.001}];
realparError = Table[{m, Chop[Re[realGateAngle]]/π}, {m, 0, 1, 0.001}];
```

In[373]:=
```
angleerrorUpPlot = ListLinePlot[{realparError, imaginaryparError},
  PlotStyle → {{Red}, {Black}},
  PlotLegends → legend[{"Re{ϵup}/π", "Im{ϵup}/π"}, {0.8, 0.35}],
  FrameLabel → {"m̃"}]
savePlot["angleerrorUpPlot.pdf", %];
Export[overleafPath <> "angleerrorUpPlot.pdf", angleerrorUpPlot];
```

Out[373]=



This plot is interesting. It informs us that in order to obtain the desired CZ-gate , i.e. -Exp[*i* ϵup]=-1, the m-parameter must be bigger than 0.4, otherwise the real part is 3π what would lead to the identity, -Exp[*i* ϵup]=1. Focusing on the part where m>0.4 we have the desired CZ-gate implemented, as the real part of ϵ_{up} is 2π. The error is unitary, if m=1 (monochromatic output) we obtain the desired gate,if 0.4<m<1 we have the imaginary part that is responsible for tilting the gate.

With this parametrization the gate can be written in a unitary form as:

In[376]:= $\texttt{Gupreal} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -e^{i\,\epsilon up} \end{pmatrix}$ // cf;

Gupideal = Gupreal /. ϵup → 2 π // mf;

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

In[378]:= (Gupreal†.Gupreal // cf) == Eye[4]

(Gupreal.Gupreal† // cf) == Eye[4]

Out[378]= True

Out[379]= True

In[380]:= Guprealsimplified = Gupreal /. ϵup → 2 π − i Log$\left[\frac{1}{2} \times (-1 + m\,(2 + m))\right]$ // cf // mf;

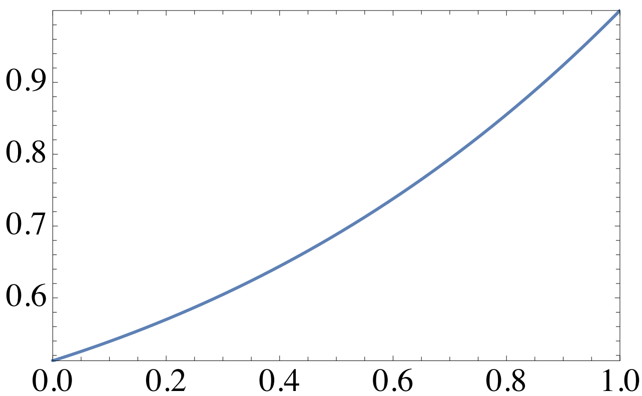
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \times (1 - m\,(2 + m)) \end{pmatrix}$$

In[381]:= averagefidelityup = Fave[Gupideal, Guprealsimplified] // cf

Plot[averagefidelityup, {m, 0, 1}]

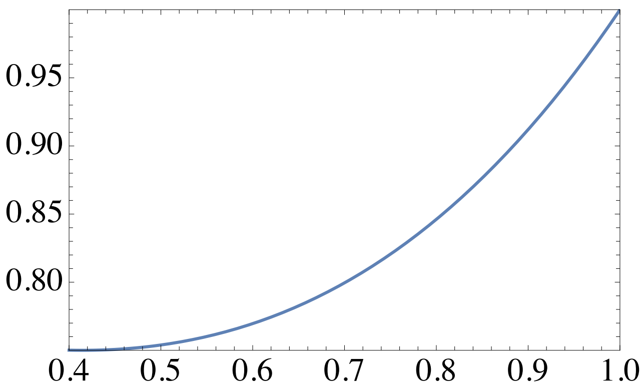
Out[381]= $\frac{1}{20} \times \left(4 + \frac{1}{4}\,(5 + m\,(2 + m))^2\right)$

Out[382]=



In[383]:= Plot[Tr[Guprealsimplified†.Guprealsimplified] / 4, {m, 0.4, 1}]

Out[383]=



The error matrix is given by:

In[384]:=
```
Eup = Gupreal.Inverse[Gupideal] // mf;
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\,\epsilon up} \end{pmatrix}$$

Decomposing in the Pauli basis (performing PTA):

In[385]:=
```
PTAup = Table[Tr[𝒫⟦i, j⟧.Eup] / 4, {i, 1, 4}, {j, 1, 4}] // cf // mf;
```

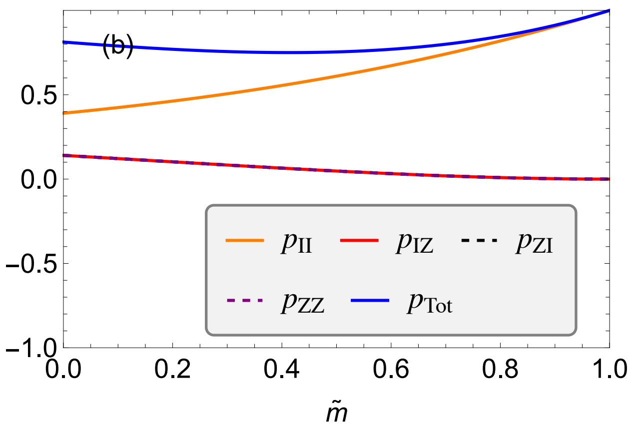
$$\begin{pmatrix} \frac{1}{4}\times\left(3+e^{i\,\epsilon up}\right) & 0 & 0 & \frac{1}{4}\times\left(1-e^{i\,\epsilon up}\right) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{4}\times\left(1-e^{i\,\epsilon up}\right) & 0 & 0 & \frac{1}{4}\times\left(-1+e^{i\,\epsilon up}\right) \end{pmatrix}$$

In[386]:=
```
PTAupm = PTAup /. ϵup → 2 π - ⅈ Log[1/2 × (-1 + m (2 + m))] // cf // mf;
```

$$\begin{pmatrix} \frac{1}{8}\times(5+m\,(2+m)) & 0 & 0 & -\frac{1}{8}\times(-1+m)\times(3+m) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{8}\times(-1+m)\times(3+m) & 0 & 0 & \frac{1}{8}\times(-1+m)\times(3+m) \end{pmatrix}$$

In[387]:=
```
Plot[{Evaluate[DeleteCases[Flatten[Abs[PTAupm]^2], 0]],
   Total[DeleteCases[Flatten[Abs[PTAupm]^2], 0]]},
  {m, 0, 1}, Frame → True, BaseStyle → 14,
  (*PlotRange→plotrangeamplitudes,*)
  FrameStyle → Black,
  ImageSize → 325,
  AxesOrigin → {0, -1},
  FrameLabel → {"m̃", None},
  PlotStyle → {Orange, {Red}, {Black, Dashed}, {Purple, Dashed}, Blue},
  PlotLegends → legend[{"pII", "pIZ", "pZI", "pZZ", "pTot"},
    {0.6, 0.23}, LegendLayout → {"Row", 2}],
  Epilog → Text["(b)", Scaled[{0.1, 0.9}]]
(*Epilog→AlphabetLabelling["c"]*)]
```

Out[387]=



Analysis for the gate conditioned on spin ↓

In[388]:=
```
GdwrealOriginal = ( 1  0  0  0
                    0 -m  0  0
                    0  0  1  0
                    0  0  0  (m^2+1)/2 ) // cf;
GdwrealOriginal /. m → 1 // mf;
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In[390]:=
```
solϵdw = Flatten@Solve[e^(i ϵdw) == m, ϵdw] // cf
```

Out[390]= $\{\epsilon dw \to 2\pi c_1 - i\,\mathrm{Log}[m] \text{ if } c_1 \in \mathbb{Z}\}$

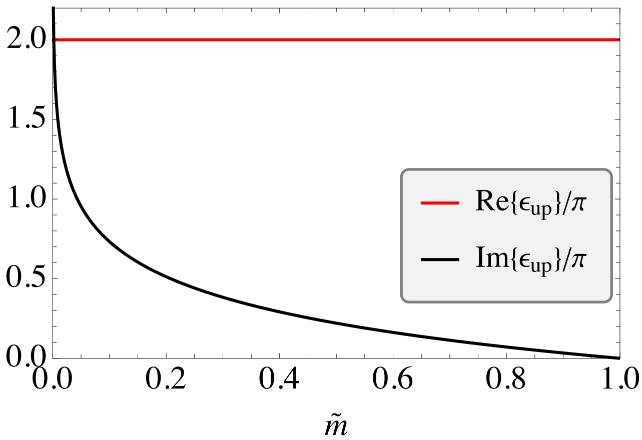
In[391]:=
```
realDownGateAngle = 2 π - i Log[m]
```

Out[391]= $2\pi - i\,\mathrm{Log}[m]$

In[392]:=
```
imaginaryparErrorDw = Table[{m, Chop[Im[realDownGateAngle]]/π}, {m, 0, 1, 0.001}];
realparErrorDw = Table[{m, Chop[Re[realDownGateAngle]]/π}, {m, 0, 1, 0.001}];
```

In[394]:=
```
angleerrorDwPlot = ListLinePlot[{realparErrorDw, imaginaryparErrorDw},
   PlotStyle → {{Red}, {Black}},
   PlotLegends → legend[{"Re{ϵ_up}/π", "Im{ϵ_up}/π"}, {0.8, 0.35}],
   FrameLabel → {"m̃"}]
Export[overleafPath <> "angleerrorDwPlot.pdf", angleerrorDwPlot];
```

Out[394]=



In[396]:=
```
solδdw = Flatten@Solve[e^(i δdw) == (m^2 + 1)/2, δdw] // cf
```

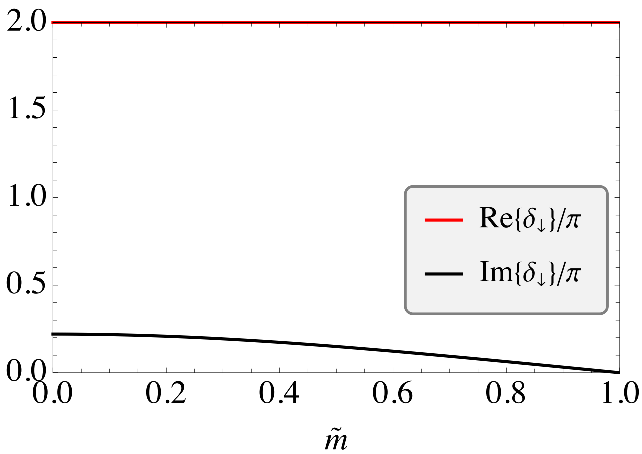
Out[396]= $\{\delta dw \to 2\pi c_1 - i\,\mathrm{Log}\left[\frac{1}{2}\times(1+m^2)\right] \text{ if } c_1 \in \mathbb{Z}\}$

In[397]:=
```
anotherUnitaryerror = 2 π - ⅈ Log[1/2 × (1 + m^2)]
```

Out[397]= $2\pi - i\,\mathrm{Log}\left[\frac{1}{2}\times(1+m^2)\right]$

In[398]:=
```
imaginaryparAnotherUnitaryErrorDw =
  Table[{m, Chop[Im[anotherUnitaryerror]]/π}, {m, 0, 1, 0.001}];
realparAnotherUnitaryErrorDw =
  Table[{m, Chop[Re[anotherUnitaryerror]]/π}, {m, 0, 1, 0.001}];
```

In[400]:=
```
anotherUnitaryerrorDwPlot = ListLinePlot[
  {realparAnotherUnitaryErrorDw, imaginaryparAnotherUnitaryErrorDw},
  PlotStyle → {{Red}, {Black}},
  PlotLegends → legend[{"Re{δ↓}/π", "Im{δ↓}/π"}, {0.8, 0.35}],
  FrameLabel → {"m̃"}]
Export[overleafPath <> "anotherUnitaryerrorDwPlot.pdf",
  anotherUnitaryerrorDwPlot];
```

Out[400]=


It can be written in a unitary form:

In[402]:=
```
realDwrotations = {ϵdw → realDownGateAngle, δdw → anotherUnitaryerror};
```

In[403]:= Gdwreal = $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -e^{i\,\epsilon dw} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\,\delta dw} \end{pmatrix}$ // cf;

```
Gdwideal = Gdwreal /. ϵdw → 0 /. δdw → 0 // mf;
Gdwreal /. realDwrotations // mf;
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -e^{i\,(2\pi - i\,\mathrm{Log}[m])} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\,\left(2\pi - i\,\mathrm{Log}\left[\frac{1}{2}\times(1+m^2)\right]\right)} \end{pmatrix}$$

In[406]:=
```
(Gdwreal†.Gdwreal // cf ) == Eye[4]
(Gdwreal.Gdwreal† // cf ) == Eye[4]
```

Out[406]= True

Out[407]= True

In[408]:=
```
averagefidelitydw = Fave[Gdwideal, (Gdwreal /. realDwrotations)] // cf
```

Out[408]= $\frac{1}{20}\times\left(4+\frac{1}{4}\,(5+m\,(2+m))^2\right)$

In[409]:=
```
averagefidelitydw - averagefidelityup // cf
```

Out[409]= 0

Error matrix down:

In[411]:=
```
Edw = Gdwreal.Inverse[Gdwideal] // mf;
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\,\epsilon dw} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\,\delta dw} \end{pmatrix}$$

In[412]:=
```
PTAdw = Table[Tr[𝒫[[i, j]].Edw] / 4, {i, 1, 4}, {j, 1, 4}] // cf // mf;
```

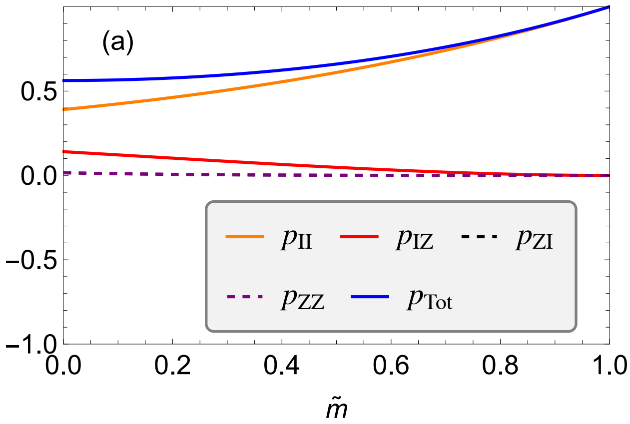
$$\begin{pmatrix} \frac{1}{4}\times\left(2+e^{i\,\delta dw}+e^{i\,\epsilon dw}\right) & 0 & 0 & \frac{1}{4}\times\left(2-e^{i\,\delta dw}-e^{i\,\epsilon dw}\right) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{4}\left(-e^{i\,\delta dw}+e^{i\,\epsilon dw}\right) & 0 & 0 & \frac{1}{4}\left(e^{i\,\delta dw}-e^{i\,\epsilon dw}\right) \end{pmatrix}$$

In[413]:=
```
PTAdwm = PTAdw /. realDwrotations // cf // mf;
```

$$\begin{pmatrix} \frac{1}{8}\times(5+m\,(2+m)) & 0 & 0 & -\frac{1}{8}\times(-1+m)\times(3+m) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{8}\,(-1+m)^2 & 0 & 0 & \frac{1}{8}\,(-1+m)^2 \end{pmatrix}$$

In[414]:=
```
Plot[{Evaluate[DeleteCases[Flatten[Abs[PTAdwm]^2], 0]],
  Total[DeleteCases[Flatten[Abs[PTAdwm]^2], 0]]},
 {m, 0, 1}, Frame → True, BaseStyle → 14,
 (*PlotRange→plotrangeamplitudes,*)
 FrameStyle → Black,
 ImageSize → 325,
 AxesOrigin → {0, -1},
 FrameLabel → {"m̃", None},
 PlotStyle → {Orange, {Red}, {Black, Dashed}, {Purple, Dashed}, Blue},
 PlotLegends → legend[{"pII", "pIZ", "pZI", "pZZ", "pTot"},
   {0.6, 0.23}, LegendLayout → {"Row", 2}],
 Epilog → Text["(a)", Scaled[{0.1, 0.9}]]]
```

Out[414]=

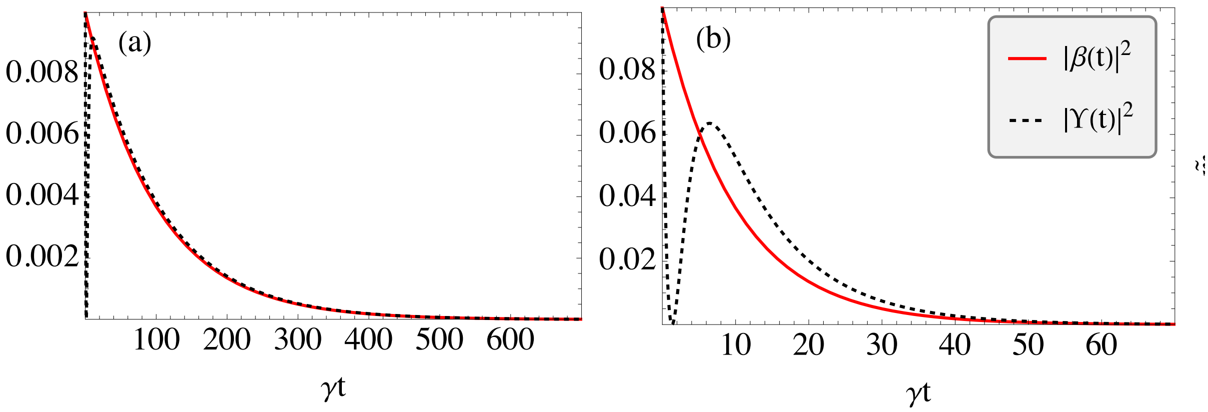


Chapter plots

Pulse temporal shape plots

In[418]:=
```
Grid[{{fig1a, fig1b, fig1c}}]
savePlot["Chap5Fig1.pdf", %];
Export[overleafPath <> "Chap5Fig1.pdf", %%];
```

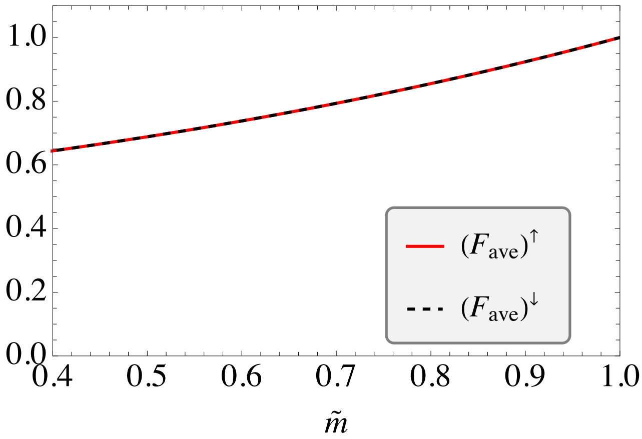
Out[418]=



Average fidelity

In[421]:=
```
Plot[{averagefidelityup, averagefidelitydw}, {m, 0, 1},
 PlotStyle → {Red, {Black, Dashed}},
 PlotRange → {{0.4, 1}, {0, 1.1}},
 PlotLegends →
  legend[{"(Fave)↑ ", "(Fave)↓"}, {0.75, 0.23}, LegendLayout → {"Row", 2}],
 FrameLabel → {"m̃", None}]
savePlot["AverageFidelities.pdf", %];
Export[overleafPath <> "AverageFidelities.pdf", %%];
```

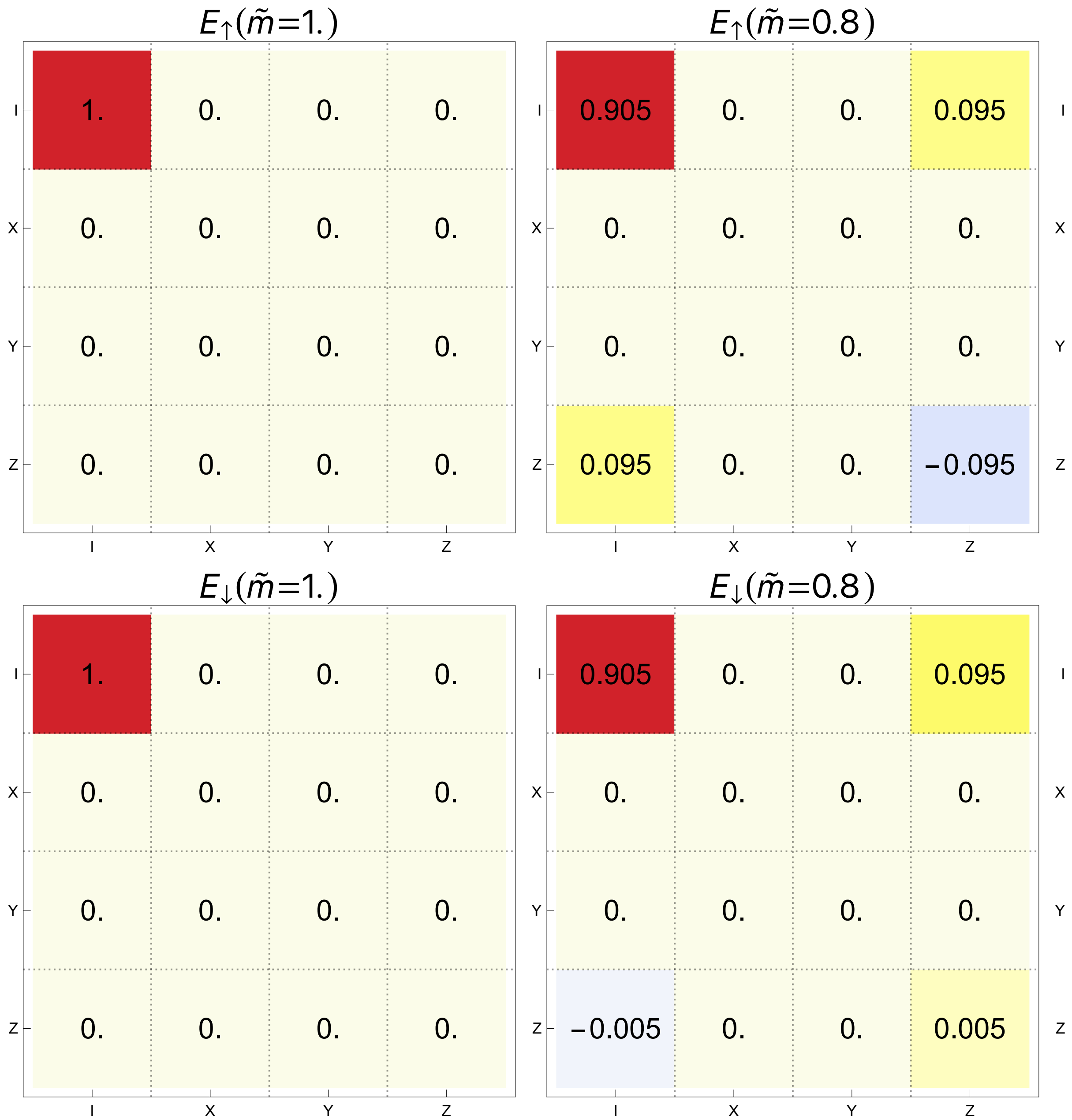
Out[421]=



Error matrix PTA

In[429]:=
```
DecompositionXYFrameTick = {{1, "I"}, {2, "X"}, {3, "Y"}, {4, "Z"}};
AllTicks = {DecompositionXYFrameTick, DecompositionXYFrameTick, None, None};
LowerRowTicks =
  {DecompositionXYFrameTick, DecompositionXYFrameTick, None, None};

Grid[{Table[MatrixPlot[PTAupm /. m → x, ColorFunctionScaling → True,
    (*Set ColorFunctionScaling to True*)ColorFunction →
     (ColorData["TemperatureMap"][Rescale[#, {-1, 1}, {-1, 1}]] &),
    ImageSize → 300, ColorFunctionScaling → True, FrameTicksStyle → Opacity[1],
    FrameTicks → AllTicks, PlotTheme → "Business",
    PlotLabel → Style["E↑(m̃=" <> ToString@x <> ")",
      Black, 20, FontFamily → "Serif"],
    Epilog → Table[Text[Style[PTAupm⟦i, j⟧ /. m → x // N, 17],
        {j - 0.5, Length[PTAupm] - i + 0.5}], {i, Length[PTAupm]},
      {j, Length[PTAupm⟦i⟧]}]], {x, {1.0, 0.8, 0.5, 0.4}}],

  Table[MatrixPlot[PTAdwm /. m → x, ColorFunctionScaling → True,
    (*Set ColorFunctionScaling to True*)ColorFunction →
     (ColorData["TemperatureMap"][Rescale[#, {-1, 1}, {-1, 1}]] &),
    ImageSize → 300, ColorFunctionScaling → True,
    FrameTicksStyle → Opacity[1], FrameTicks → LowerRowTicks,
    PlotTheme → "Business",
    PlotLabel →
     Style["E↓(m̃=" <> ToString@x <> ")", Black, 20, FontFamily → "Serif"],
    Epilog → Table[Text[Style[PTAdwm⟦i, j⟧ /. m → x // N, 17],
        {j - 0.5, Length[PTAdwm] - i + 0.5}], {i, Length[PTAdwm]},
      {j, Length[PTAdwm⟦i⟧]}]], {x, {1.0, 0.8, 0.5, 0.4}}]}]
savePlot["HintonDiagramErrorPTA.pdf", %];
Export[overleafPath <> "HintonDiagramErrorPTA.pdf", %%];
```

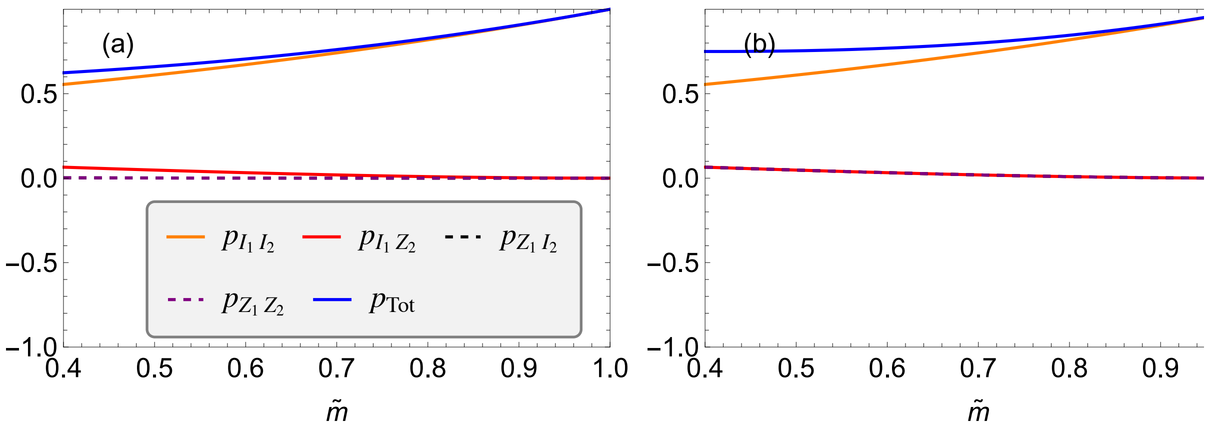
Out[432]=


$E_{\uparrow}(\tilde{m}=1.)$
1. 0. 0. 0.
0. 0. 0. 0.
0. 0. 0. 0.
0. 0. 0. 0.
I X Y Z
$E_{\uparrow}(\tilde{m}=0.8)$
0.905 0. 0. 0.095
0. 0. 0. 0.
0. 0. 0. 0.
0.095 0. 0. −0.095
I X Y Z
$E_{\downarrow}(\tilde{m}=1.)$
1. 0. 0. 0.
0. 0. 0. 0.
0. 0. 0. 0.
0. 0. 0. 0.
I X Y Z
$E_{\downarrow}(\tilde{m}=0.8)$
0.905 0. 0. 0.095
0. 0. 0. 0.
0. 0. 0. 0.
−0.005 0. 0. 0.005
I X Y Z

Non-null components of PTA

In[438]:=
```
Grid[{{Plot[{Evaluate[DeleteCases[Flatten[Abs[PTAdwm]^2], 0]],
    Total[DeleteCases[Flatten[Abs[PTAdwm]^2], 0]]},
   {m, 0.4, 1}, Frame → True, BaseStyle → 14,
   (*PlotRange→plotrangeamplitudes,*)
   FrameStyle → Black,
   ImageSize → 325,
   AxesOrigin → {0.4, -1},
   FrameLabel → {"m̃", None},
   PlotStyle → {Orange, {Red}, {Black, Dashed}, {Purple, Dashed}, Blue},
   PlotLegends → legend[{"p_{I1 I2}", "p_{I1 Z2}", "p_{Z1 I2}", "p_{Z1 Z2}", "p_Tot"},
     {0.55, 0.23}, LegendLayout → {"Row", 2}],
   Epilog → Text["(a)", Scaled[{0.1, 0.9}]]],

  Plot[{Evaluate[DeleteCases[Flatten[Abs[PTAupm]^2], 0]],
    Total[DeleteCases[Flatten[Abs[PTAupm]^2], 0]]},
   {m, 0.4, 1}, Frame → True, BaseStyle → 14,
   (*PlotRange→plotrangeamplitudes,*)
   FrameStyle → Black,
   ImageSize → 325,
   AxesOrigin → {0.4, -1},
   FrameLabel → {"m̃", None},
   PlotStyle → {Orange, {Red}, {Black, Dashed}, {Purple, Dashed}, Blue},
   (*PlotLegends→
    legend[{"e_II","e_IZ","e_ZI","e_ZZ"},{0.7,0.23},LegendLayout→{"Row",2}],*)
   Epilog → Text["(b)", Scaled[{0.1, 0.9}]]
   (*Epilog→AlphabetLabelling["c"]*)]}}
]
savePlot["NonVanishingComponentsPTA.pdf", %];
Export[overleafPath <> "NonVanishingComponentsPTA.pdf", %%];
```

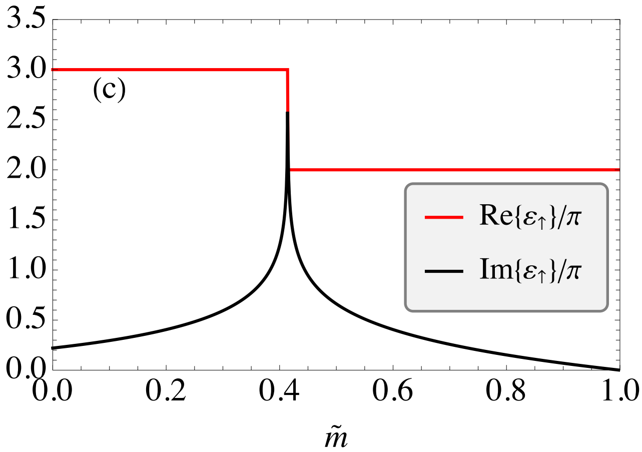
Out[438]=



Real and imaginary parts of the real angles

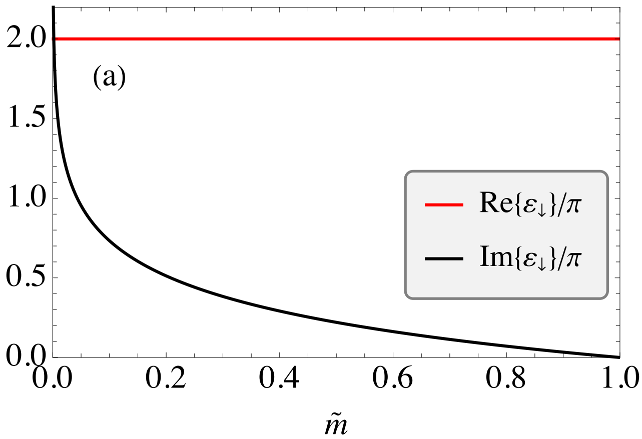
In[441]:=
```
angleerrorUpPlot = ListLinePlot[{realparError, imaginaryparError},
   PlotStyle → {{Red}, {Black}},
   PlotRange → {All, {0, 3.5}},
   PlotLegends → legend[{"Re{ε↑}/π", "Im{ε↑}/π"}, {0.8, 0.35}],
   FrameLabel → {"m̃", None},
   Epilog → Text["(c)", Scaled[{0.1, 0.8}]]]
savePlot["angleerrorUpPlot.pdf", angleerrorUpPlot];
Export[overleafPath <> "angleerrorUpPlot.pdf", angleerrorUpPlot];
```

Out[441]=

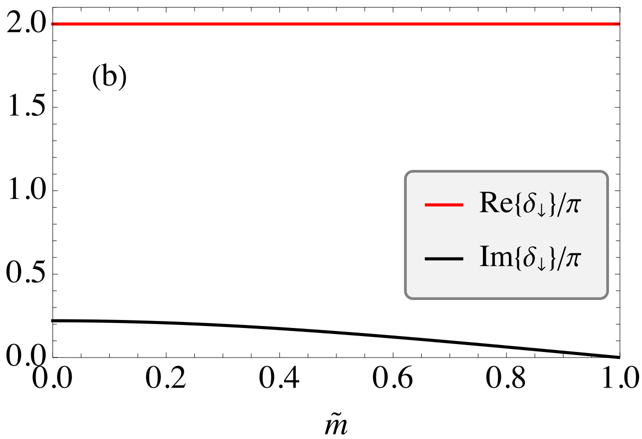


In[444]:=
```
angleerrorDwPlot = ListLinePlot[{realparErrorDw, imaginaryparErrorDw},
  PlotStyle → {{Red}, {Black}},
  PlotLegends → legend[{"Re{ε↓}/π", "Im{ε↓}/π"}, {0.8, 0.35}],
  FrameLabel → {"m̃", None},
  Epilog → Text["(a)", Scaled[{0.1, 0.8}]]]
anotherUnitaryerrorDwPlot = ListLinePlot[
  {realparAnotherUnitaryErrorDw, imaginaryparAnotherUnitaryErrorDw},
  PlotStyle → {{Red}, {Black}},
  PlotRange → {All, {0, 2.1}},
  PlotLegends → legend[{"Re{δ↓}/π", "Im{δ↓}/π"}, {0.8, 0.35}],
  FrameLabel → {"m̃", None},
  Epilog → Text["(b)", Scaled[{0.1, 0.8}]]]
savePlot["anotherUnitaryerrorDwPlot.pdf", anotherUnitaryerrorDwPlot];
Export[overleafPath <> "anotherUnitaryerrorDwPlot.pdf",
  anotherUnitaryerrorDwPlot];
```

Out[444]=

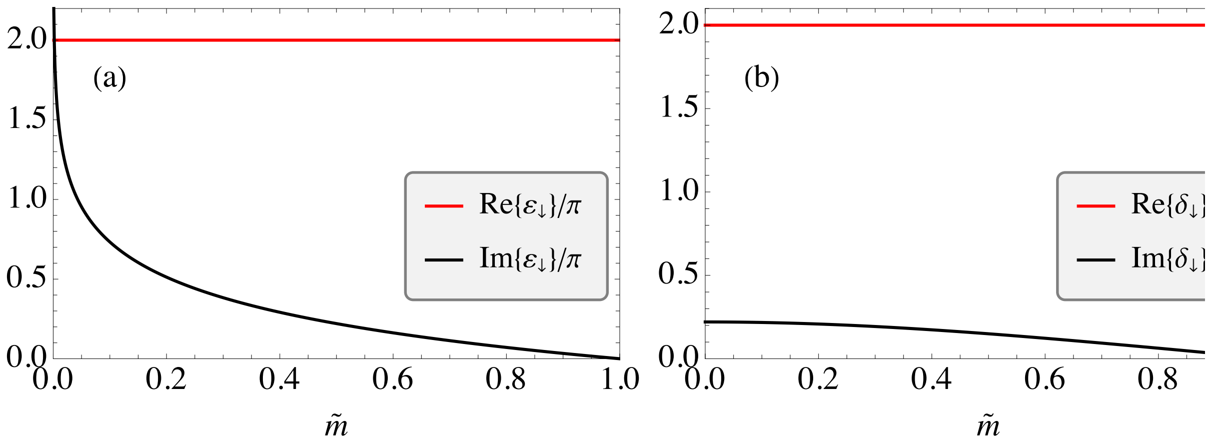


Out[445]=



In[448]:=
```
Grid[{{angleerrorDwPlot, anotherUnitaryerrorDwPlot, angleerrorUpPlot}}]
savePlot["GatesErrorsRealandImaginaryParts.pdf", %];
Export[overleafPath <> "GatesErrorsRealandImaginaryParts.pdf", %%];
```

Out[448]=



In Figure (a), we observe that the real part of $\varepsilon\downarrow\varepsilon\downarrow$ remains constant at $2\pi2\pi$ throughout, while the imaginary part emerges as mm approaches 0, leading to an induced error. In Figure (b), the real part of $\delta\downarrow\delta\downarrow$ also remains constant at $2\pi2\pi$ for all mm, and the imaginary part appears with a much smaller magnitude than $2\pi2\pi$ as mm approaches 0, becoming a constant error for m<0.4m<0.4. In Figure (c), we find that to achieve the desired CZ-gate, i.e., –Exp[iε↑]=–1–Exp[iε↑]=–1, the mm-parameter must exceed 0.4; otherwise, the real part becomes $3\pi3\pi$, resulting in the identity gate, –Exp[iε↑]=1–Exp[iε↑]=1. Focusing on the region where m>0.4m>0.4, we successfully implement the desired CZ-gate, as the real part of $\varepsilon\uparrow\varepsilon\uparrow$ is $2\pi2\pi$. The error is unitary when m=1m=1 (monochromatic output), yielding the desired gate. For 0.4<m<10.4<m<1, the imaginary part comes into play, causing a tilt in the gate.

In[451]:=
```
leakageDw = Tr[GdwrealOriginal†.GdwrealOriginal] / 4 // cf
leakageUp = Tr[GuprealOriginal†.GuprealOriginal] / 4 // cf
```

Out[451]= $\frac{1}{16}\left(3+m^2\right)^2$

Out[452]= $\frac{1}{16}\times(13+m\,(2+m)\times(-2+m\,(2+m)))$

In[453]:=
```
leakagePlots = Plot[{leakageDw, Total[DeleteCases[Flatten[Abs[PTAdwm]^2], 0]],
    leakageUp, Total[DeleteCases[Flatten[Abs[PTAupm]^2], 0]]}, {m, 0, 1},
   PlotStyle → {Black, {Red, Dashed}, Orange, {Blue, Dashed}},
   PlotRange → {All, {0, 1.1}},
   PlotLegends → legend[{"Leak↓", "(pTot)↓", "Leak↑", "(pTot)↑"}, {0.8, 0.35}],
   FrameLabel → {"m̃", None},
   Epilog → {{Gray}, Line[{{0, 1}, {1, 1}}]}]
savePlot["Non-unitarity.pdf", %];
Export[overleafPath <> "Non-unitarity.pdf", %];
```

Out[453]=