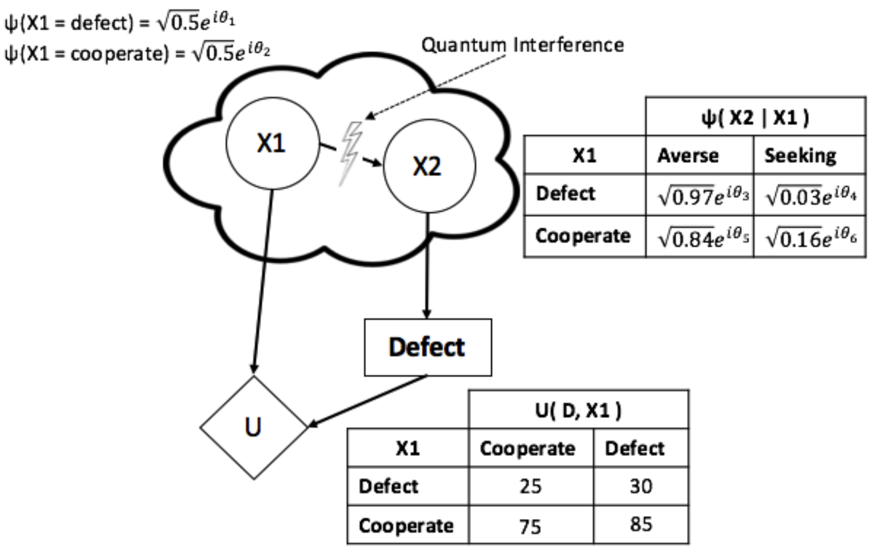
Author: Catarina Moreira
Email: catarina.pintomoreira@qut.edu.au
Institution: School of Information Systems, Queensland University of Technology
Date of Last version: 24 / 01 / 2020

Quantum-Like Influence Diagrams for Violations of the Sure Thing Principle

Introduction

The general idea is to take advantage of the quantum interference terms produced in the quantum-like Bayesian Network to influence the probabilities used to compute the expected utility of some action. This way, we are not proposing a new type of expected utility hypothesis. On the contrary, we are keeping it under its classical definition. We are only incorporating it as an extension of a probabilistic graphical model in a compact graphical representation called an influence diagram in which the utility function depends on the probabilistic influences of the quantum-like Bayesian network.

Illustrative Problem: Decision Scenario in Prisoner's Dilemma Game (Tversky & Shafir(1992))



	ψ(X2 | X1)	
X1	Averse	Seeking
Defect	$\sqrt{0.97}e^{i\theta_3}$	$\sqrt{0.03}e^{i\theta_4}$
Cooperate	$\sqrt{0.84}e^{i\theta_5}$	$\sqrt{0.16}e^{i\theta_6}$

	U(D, X1)	
X1	Cooperate	Defect
Defect	25	30
Cooperate	75	85

Classical Definition

We can use the ideas in the context of probabilistic graphical models to represent the Decision-Making situation in a very intuitive and interpretable way though influence diagrams.
Very generally, a decision-making scenario D contains the following attributes:

- A set of possible actions Val(A) = { A^1, A^2, ..., A^N } which correspond to choices that an agent can choose (e.g. drugs that can be given to a patient).
- A set of possible states Val(X) = { X^1, X^2, ..., X^N } which correspond to the states of the world that the agent can (and cannot) affect (eg. the effects that drug A had in some patient)
- A probability distribution Pr(X | A) that represents the probability distribution of obtaining a certain state, given that the agent chose some action.
- An utility function U(X,A) which defines the agent's preferences. It allows the measurement of the agent's satisfaction when choosing different actions in different states

This can be formalised using the notion of Maximum Expected Utility, MEU.
Given an a decision problem, D, and a set of possible actions, the goal is to choose action *a that maximizes the Expected Utility formula:*

$$\mathrm{Eu}[D[\delta_A]] = \sum_{x \in X} \sum_{a \in A} \mathrm{Pr}_{\delta A}(x \mid a)\, U(x, a) \quad \text{where we choose} \quad a^* = \mathrm{argmax}_{\delta_A} \mathrm{Eu}[D[\delta_A]]$$

$$\text{Maximum Expected Utiltiy}: \quad \mathrm{MEU}(D) = \max_{\delta_A} \mathrm{Eu}[D[\delta_A]]$$

$$\delta_D{}^* = \begin{cases} 1 & \mathrm{argmax}_{\delta_D} \mathrm{Eu}[D[\delta_A]] \\ 0 & \text{otherwise} \end{cases}$$

Finding MEU Rules for Classical Decision-Making

For the particular example in Figure 1, we can rewrite the Expected Utility in the following way:

$$\mathrm{Eu}[D[\delta_A]] = \sum_{x \in X} \sum_{a \in A} \mathrm{Pr}_{\delta A}(x \mid a)\, U(x, a)$$

$$\mathrm{Eu}[D[\delta_A]] = \sum_{X1, X2, D} \mathrm{Pr}(X1)\, \mathrm{Pr}(X2 \mid X1)\, \delta_D(D \mid X2)\, U(X1, D)$$

Since this represents a product of factors, we can separate the factors that we do not depend on the optimization rule

$$= \sum_{X2,\, D} \delta_D(D \mid X2) \sum_{X1} \mathrm{Pr}(X1)\, \mathrm{Pr}(X2 \mid X1)\, \delta_D(D \mid X2)\, U(X1, D) \tag{1}$$

The resulting factor, will be defined as μ(X2,D),

$$\mathrm{Eu}[D[\delta_A]] = \sum_{X2,\, D} \delta_D(D \mid X2)\, \mu(X2, D) \tag{2}$$

However, since the above formula obeys to the axioms of expected utility theory, influence diagrams cannot take into account human paradoxical decisions under uncertainty, such as violations to the sure thing principle.

Quantum-Like Influence Diagrams

A Quantum-Like Influence Diagram is a compact directed acyclical graphical representation of a decision scenario, which was originally proposed by Howard and Matheson (1984). It consists on a set of random variables $X_1, ..., X_N$ belonging to a quantum-like Bayesian network. Each random variable X_i is associated with a conditional probability distribution (CPD) table, which describes the distribution of quantum probability amplitudes of the random variable X_i with respect to its parent nodes, $\psi(X_i \mid Pa_{X_i})$. Note that the difference between a quantum-like Bayesian network and a classical network is simply the usage of complex numbers instead of classical real numbers. The usage of complex numbers will enable the emergence of quantum interference effects. The influence diagram also consists in an utility node defined variable U, which is associated with a deterministic function $U(Pa_U)$. The goal is to make a decision, which maximises the expected utility function by taking into account probabilistic inferences performed on the quantum-like Bayesian network.

Finding MEU Rules for Quantum-Like Decision-Making

The goal is to use quantum-like probabilistic inferences that will influence the Maximum Expected Utility and that will allow us to find decision rules that can accommodate violations to the Sure Thing Principle. We do this, under the formalism of the Quantum-Like Bayesian Networks, where we define each random variable as quantum states in a complex Hilbert Space

For the particular example in Figure 1, we can rewrite the Expected Utility in the following way:

$$\text{Eu}[D[\delta_A]] = \sum_{x \in X} \sum_{a \in A} \psi_{\delta A}(x \mid a)\, U(x, a)$$
$$\text{Eu}[D[\delta_A]] = \sum_{X1, X2, D} \psi(X1)\, \psi(X2 \mid X1)\, \delta_D(D \mid X2)\, U(X1, D) \quad (3)$$

Since this represents a product of factors, we can separate the factors that we do not depend on the optimization rule

$$= \sum_{X2,\, D} \delta_D(D \mid X2) \left| \sum_{X1} \psi(X1)\, \psi(X2 \mid X1)\, \psi(D \mid X2) \right|^2 \sum_{X1} U(X1, D) \quad (4)$$

The marginalisation will produce two terms: one that corresponds to classical probability and another that corresponds to quantum interference effects:

$$= \sum_{X2,\, D} \delta_D(D \mid X2)\, (\Pr(X2) + \text{interferece}) \sum_{X1} U(X1, D) \quad (5)$$

This will lead to a factor containing the decision that the Player needs to make (either to *defect* or *cooperate*) and two utility factors

- One utility factor that corresponds to the classical expected utility theory: μ(X2,D)
- One utility factor that corresponds to the influence of interference effects in the utility function: π(X2,D)

In the end, the Maximum Expected Utility under quantum-like influence diagrams is given by:

$$\text{Eu}[D[\delta_A]] = \sum_{X2,\, D} \delta_D(D \mid X2)\, \mu(X2, D)\, \pi(X2, D) \quad \text{where we choose} \quad a^* =$$

$\text{argmax}_{\delta_D}\, Eu[D[\delta_D]]$

Maximum Expected Utiltiy : $\quad MEU\,(D) = \max_{\delta_D} Eu[D[\delta_D]]$

$$\delta_D{}^{*}\,(D \mid X2) = \begin{cases} 1 & \text{argmax}_{\delta_D}\, \mu\,(X2, D)\, \pi\,(X2, D) \\ 0 & \text{otherwise} \end{cases}$$

Evaluation of the Model in the Prisoner’s Dilemma Game over Several Different Experiments of the Literature with Different Payoffs.

In this section, we apply the formalisms of quantum-like influence diagrams for the following works of the literature:

- Tversky & Shafir (1992)
- Li & Taplin (2002) - seven different games were made with different payoffs

Literature	**Known to Defect**	**Known to Collaborate**	**Unknown**	**Clas**
Shafir and Tversky (1992)	0.9700	0.8400	0.6300	
Li and Taplin (2002) (Average)	0.8200	0.7700	0.7200	
Li and Taplin (2002) **Game 1**	0.7333	0.6670	0.6000	
Li and Taplin (2002) **Game 2**	0.8000	0.7667	0.6300	
Li and Taplin (2002) **Game 3**	**0.9000**	**0.8667**	**0.8667**	
Li and Taplin (2002) **Game 4**	0.8333	0.8000	0.7000	
Li and Taplin (2002) **Game 5**	0.8333	0.7333	0.7000	
Li and Taplin (2002) **Game 6**	**0.7667**	**0.8333**	**0.8000**	
Li and Taplin (2002) **Game 7**	**0.8667**	**0.7333**	**0.7667**	

Table 1. Works of the literature reporting the probability of a player choosing to de conditions. The entries of the table that are highlighted correspond to experiments wl of the sure thing principle were not found.

The corresponding payoffs are (we should change the sign, because we want to maximize these values):

	Shafir and Tversky (1992)		Li and Taplin (2002) Game 1		Game 2		Game 3		Game 4		Game 5		Game 6		Game 7	
Payoff																
dd dc	30	25	30	25	73	25	30	25	80	78	43	10	30	10	30	10
cd cc	85	75	85	75	85	75	85	36	85	83	85	46	60	33	60	33

In[9]:=
```
Clear["Global`*"]
```

In[7]:=
```
(* ############## DEFINING PAYOFFS AND
 UTILITIES ################################ *)
(* we define the player's payoffs according to Table 2 *)
```

```
In[10]:= dd = 30;  (* when both players defect *)
         dc = 25;  (* when Player 1 defects and Player 2 cooperates *)
         cd = 85;  (* when Player 1 cooperates and Player 2 defects *)
         cc = 75;  (* when both players cooperate *)

         (* ############### DEFINING PROBABILITY DISTRIBUTIONS
          FROM EXPERIMENTAL DATA #################################### *)
         (* the conditional probability tables of the quantum-
          like Bayesian network are given by psychological experimental
           findings from the literature, which are presented in Table 1 *)

In[16]:= pdd = 0.97;
         pcd = 0.84;
         pdc = 1 - pdd;
         pcc = 1 - pcd;

         (* ############### DEFINING QUANTUM STATES ################################### *)
         (* Each classical random variable corresponds to a Quantum State *)
         (* Define the Quantum State |X1〉 correspoding
          to the 1st player of the Prisoner's Dilemma *)
         (* |X1〉 = α1|X1=defect〉 + α2|X1=cooperate〉 *)

In[20]:= X1 = ( Sqrt[0.5] );
              ( Sqrt[0.5] )

         (* Define the Quantum Conditional Probability
          States |X2X1d〉, |X2X1c〉. Each state is defined in its own Hilbert Space *)

         (*|X2X1〉: Player 2 Prob. Amplitudes,
         given Player 1 is believed to have chosen Defect *)
         (*        Player 2 Prob. Amplitudes given
          Player 1 is believed to have chosen Cooperate *)

In[21]:= X2X1 = ( Sqrt[pdd] ⅇ^(ⅈ Re[θ1])  Sqrt[pcd] ⅇ^(ⅈ Re[θ3]) );
                ( Sqrt[pdc] ⅇ^(ⅈ Re[θ2])  Sqrt[pcc] ⅇ^(ⅈ Re[θ4]) )

         (* ########## COMPUTE THE FULL JOINT PROBABILITY DISTRIBUTION ########## *)
         (* The goal is to create a quantum density matrix
           that corresponds to the product of the factors of the
           quantum states that represent the random variables of the QBN
            Joint = ∑^2_{i=1}∑^2_{j=1}∑^2_{k=1} αi βj1 βk2 |X1 = i ⊗ X2X1d = j ⊗ X2X1c = k 〉 *)

In[22]:= J = ConstantArray[0, {4}];
         For[c = 1; i = 1, i ≤ 2, i++,
           For[j = 1, j ≤ 2, j++,
            J[[c]] = X1[[i]] X2X1[[j, i]];
            c = c + 1;
           ]];

         (* Simplify Joint generate density matrix from
          product of factors (or product of quantum states *)

         (* Joint = |J〉〈J| *)
```

```
In[24]:= Joint = FullSimplify[J . ConjugateTranspose[J]];

In[25]:= MatrixForm[Joint]
```

Out[25]//MatrixForm=

$$\begin{pmatrix} 0.485 & 0.0852936\, e^{i\,\mathrm{Re}[\theta1-\theta2]} & 0.451331\, e^{i\,\mathrm{Re}[\theta1-\theta3]} & 0.196977\, e^{i\,\mathrm{Re}[\theta1-\theta4]} \\ 0.0852936\, e^{-i\,\mathrm{Re}[\theta1-\theta2]} & 0.015 & 0.0793725\, e^{i\,\mathrm{Re}[\theta2-\theta3]} & 0.034641\, e^{i\,\mathrm{Re}[\theta2-\theta4]} \\ 0.451331\, e^{-i\,\mathrm{Re}[\theta1-\theta3]} & 0.0793725\, e^{-i\,\mathrm{Re}[\theta2-\theta3]} & 0.42 & 0.183303\, e^{i\,\mathrm{Re}[\theta3-\theta4]} \\ 0.196977\, e^{-i\,\mathrm{Re}[\theta1-\theta4]} & 0.034641\, e^{-i\,\mathrm{Re}[\theta2-\theta4]} & 0.183303\, e^{-i\,\mathrm{Re}[\theta3-\theta4]} & 0.08 \end{pmatrix}$$

```
(* Check if calculations are correct. The
 classical full joint distribution should correspond to
 the sum of all diagonal elements of the density matrix *)

In[26]:= If[Total[Diagonal[Joint]] == 1,
 Print[Correct!],
 Print[Error!]]

Correct!

(* ################## Generate Quantum-Like Inferences ######################## *)
(* The goal is to create an operator that selects the entries
 in the density matrix that correspond to a specifc query *)
(* OPERATORS: *)
(* X1 X2 | Pr( X1, X2 ) *)
(* ------------------------------ *)
(* D  D  | Pr(X1=D) Pr(X2=D|X1=D) *)
(* D  C  | Pr(X1=D) Pr(X2=C|X1=C) *)
(* C  D  | Pr(X1=C) Pr(X2=D|X1=D) *)
(* C  C  | Pr(X1=C) Pr(X2=D|X1=C) *)

(* DEFECT: computed just like in the classical
  setting. Select the 1st and 3rd position of the classical joint *)
```

In[27]:= Defect = $\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$;

```
(* COOPERATE: computed just like in the classical
  setting. Select the 2nd and 4th position of the classical joint *)
```

In[28]:= Cooperate = $\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$;

```
(* CLASSICAL: *)

(* Compute Pr( P2 = defect ) = Trace[ Defect' Joint Defect ] = 0.905 *)

In[29]:= P2DefClassical = (Transpose[Defect].Diagonal[Joint])[[1]]

Out[29]= 0.905

(* Compute Pr( P2 = cooperate ) = Trace[ Cooperate Joint ] = 0.095 *)
```

In[30]:=
```
P2CoopClassical = (Transpose[Cooperate].Diagonal[Joint ])[[1]]
```
Out[30]=
```
0.095
```

```
(* QUANTUM: *)

(* PARTIAL TRACE FUCNTION *)

(* Compute Pr( P2 = defect ) = α|Defect Joint|^2 = 0.905 + Int. *)

(* where α is a normalization factor that
 corresponds to: Pr( P2 = defect ) + Pr( P2 = coop ) *)
```

In[33]:=
```
P2DefQuantum =
 FullSimplify[Transpose[Defect] . Joint. Defect, θDef == θ1 - θ3][[1]][[1]]
```
Out[33]=
```
0.905 + 0.902663 Cos[Re[θDef]]
```

```
(* Compute Pr( P2 = cooperate ) = |Cooperate Joint|^2 = 0.095 + Int. *)
```

In[34]:=
```
P2CoopQuantum =
 FullSimplify[Transpose[Cooperate] . Joint. Cooperate, θCoop == θ2 - θ4][[1]][[1]]
```
Out[34]=
```
0.095 + 0.069282 Cos[Re[θCoop]]
```

```
(* Normalize Results *)
```

In[35]:=
```
Normalization[x_, y_] := x / (x + y)
```

In[38]:=
```
(* Quantum probability of defecting *)
```

In[36]:=
```
P2DefQuantumNorm = Normalization[P2DefQuantum, P2CoopQuantum]
```
Out[36]=

$$\frac{0.905 + 0.902663\, \mathrm{Cos}[\mathrm{Re}[\theta\mathrm{Def}]]}{1. + 0.069282\, \mathrm{Cos}[\mathrm{Re}[\theta\mathrm{Coop}]] + 0.902663\, \mathrm{Cos}[\mathrm{Re}[\theta\mathrm{Def}]]}$$

In[39]:=
```
(* Quantum probability of cooperating *)
```

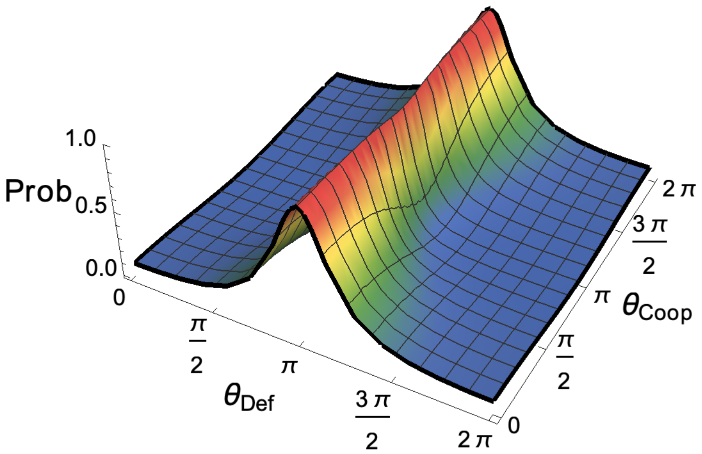
In[37]:=
```
P2CoopQuantumNorm = Normalization[P2CoopQuantum, P2DefQuantum]
```
Out[37]=

$$\frac{0.095 + 0.069282\, \mathrm{Cos}[\mathrm{Re}[\theta\mathrm{Coop}]]}{1. + 0.069282\, \mathrm{Cos}[\mathrm{Re}[\theta\mathrm{Coop}]] + 0.902663\, \mathrm{Cos}[\mathrm{Re}[\theta\mathrm{Def}]]}$$

In[40]:=
```
(* ########## QUANTUM PROBABILITY OF COOPERATING
 ACCORDING TO DIFFERENT INTERFERENCES ########### *)
```

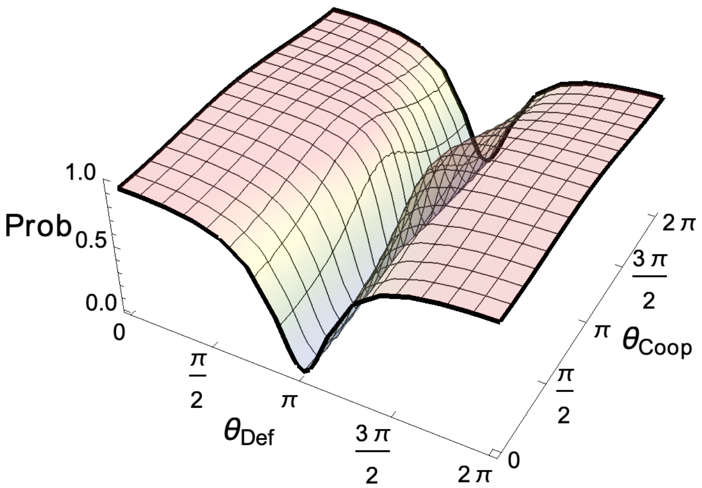
```
In[41]:= qProbCoop = Plot3D[P2CoopQuantumNorm, {θDef, 0, 2 π},
    {θCoop, 0, 2 π}, ColorFunction → (ColorData["DarkRainbow"][#3] &),
    AxesLabel → {Style["θDef", 16], Style["θCoop", 16], Style["Prob", 16]},
    BoundaryStyle → Thick, Boxed → False,
    Ticks → {{0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, {0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, Automatic},
    TicksStyle → Directive[Black, 12]]
```

Out[41]=




```
(* ########## QUANTUM PROBABILITY OF DEFECTING
 ACCORDING TO DIFFERENT INTERFERENCES ########### *)

In[42]:= qProbDef = Plot3D[P2DefQuantumNorm, {θDef, 0, 2 π},
    {θCoop, 0, 2 π}, ColorFunction → (ColorData["DarkRainbow"][#3] &),
    AxesLabel → {Style["θDef", 16], Style["θCoop", 16], Style["Prob", 16]},
    BoundaryStyle → Thick, Boxed → False,
    Ticks → {{0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, {0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, Automatic},
    TicksStyle → Directive[Black, 12], PlotStyle → Directive[Opacity[0.2], Red]]
```

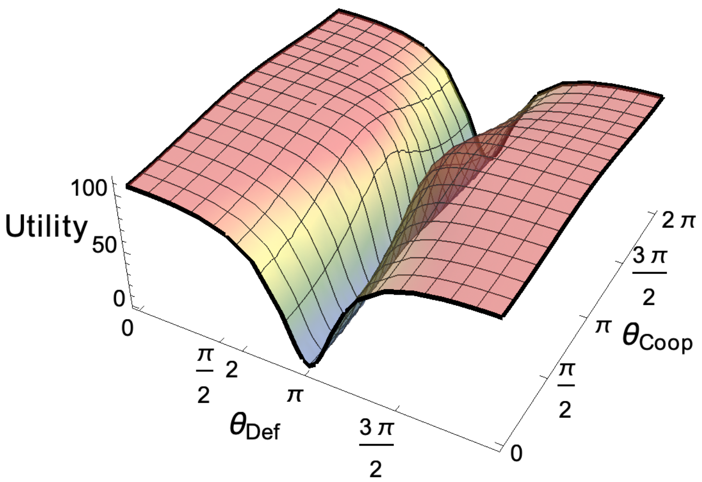
Out[42]=




```
(* ####################### DEFINING UTILITY STATES ######################### *)

(* Define the Utility node, U, as a matrix with the payoffs of the game *)
(* Player X2 wins 85 if he chooses defect when X1 also chose defect *)
(* Player X2 wins 85 if he chooses cooperate when X1 also chose defect *)
```

```
In[43]:= Udef = ( dd 0  0  0
                  0  0  0  0
                  0  0  cd 0
                  0  0  0  0 );

In[44]:= UtilityX2d = Simplify[Total[Diagonal[P2DefQuantumNorm Udef]]]

                    115.298 + 115. Cos[Re[θDef]]
Out[44]= ─────────────────────────────────────────────────────
         1.10783 + 0.076753 Cos[Re[θCoop]] + 1. Cos[Re[θDef]]

In[53]:= (* ########## EXPECTED UTILITY OF DEFECTING ACCORDING TO THE QUANTUM-
          LIKE INFLUENCE DIAGRAMS ########### *)

In[45]:= utilDef =
          Plot3D[UtilityX2d, {θDef, 0, 2 π}, {θCoop, 0, 2 π}, ColorFunction → "DarkRainbow",
           AxesLabel → {Style["θDef", 16], Style["θCoop", 16], Style["Utility", 16]},
           BoundaryStyle → Thick, Boxed → False,
           Ticks → {{0, Pi/2, Pi, 3 Pi/2, 2, Pi}, {0, Pi/2, Pi, 3 Pi/2, 2 Pi}, Automatic},
           TicksStyle → Directive[Black, 12], PlotStyle → Directive[Opacity[0.5], Red]]
```

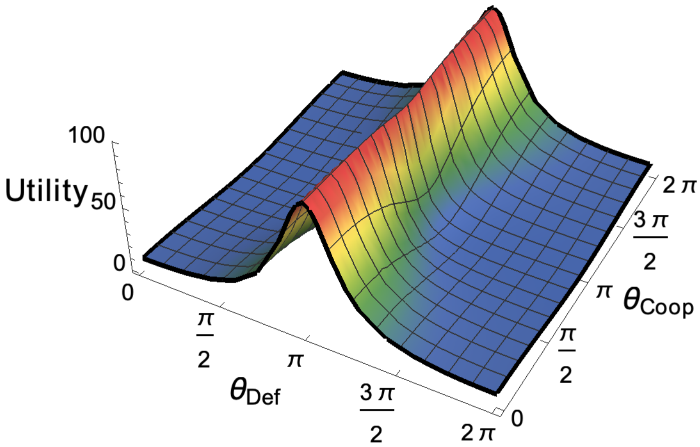
Out[45]=




```
(* Player X2 wins 25 if he chooses defect when X1 also chose cooperate *)
(* Player X2 wins 75 if he chooses cooperate when X1 also chose cooperate *)

In[47]:= Ucoop = ( 0  0  0  0
                   0  dc 0  0
                   0  0  0  0
                   0  0  0  cc );

In[49]:= UtilityX2c = FullSimplify[Total[Diagonal[P2CoopQuantumNorm Ucoop]]]

                    137.121 + 100. Cos[Re[θCoop]]
Out[49]= ─────────────────────────────────────────────────────
         14.4338 + 1. Cos[Re[θCoop]] + 13.0288 Cos[Re[θDef]]
```

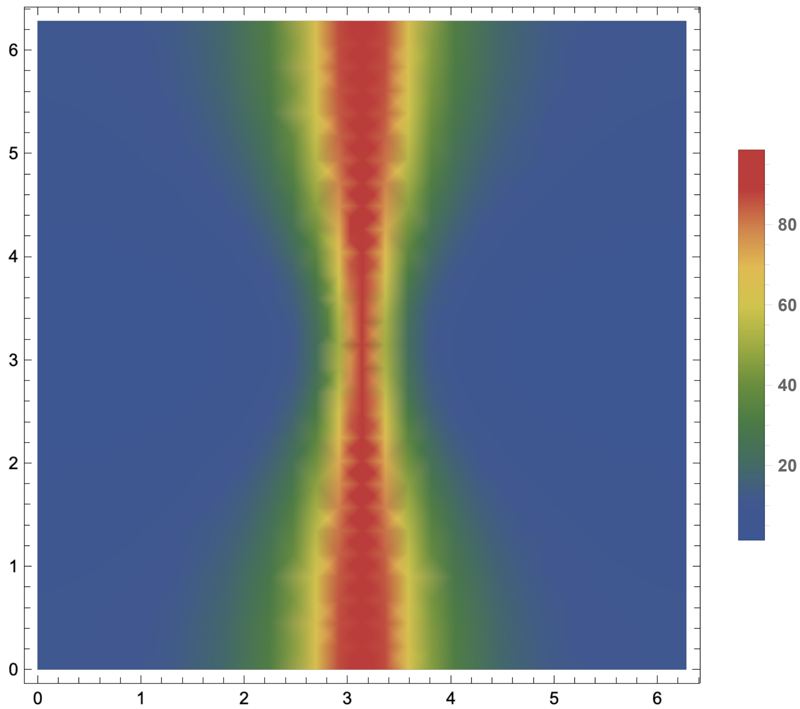
```
In[50]:= utilCoop = Plot3D[UtilityX2c, {θDef, 0, 2 π},
   {θCoop, 0, 2 π}, ColorFunction → (ColorData["DarkRainbow"][#3] &),
   AxesLabel → {Style["θDef", 16], Style["θCoop", 16], Style["Utility", 16]},
   BoundaryStyle → Thick, Boxed → False,
   Ticks → {{0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, {0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, Automatic},
   TicksStyle → Directive[Black, 12]]
```

Out[50]=




```
In[54]:= (* ########## EXPECTED UTILITY OF COOPERATION ACCORDING TO THE QUANTUM-
  LIKE INFLUENCE DIAGRAMS ########### *)

In[52]:= DensityPlot[UtilityX2c, {θDef, 0, 2 π}, {θCoop, 0, 2 π},
  Ticks → {{0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, {0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}},
  ColorFunction → "DarkRainbow", PlotLegends → Automatic]
```

Out[52]=

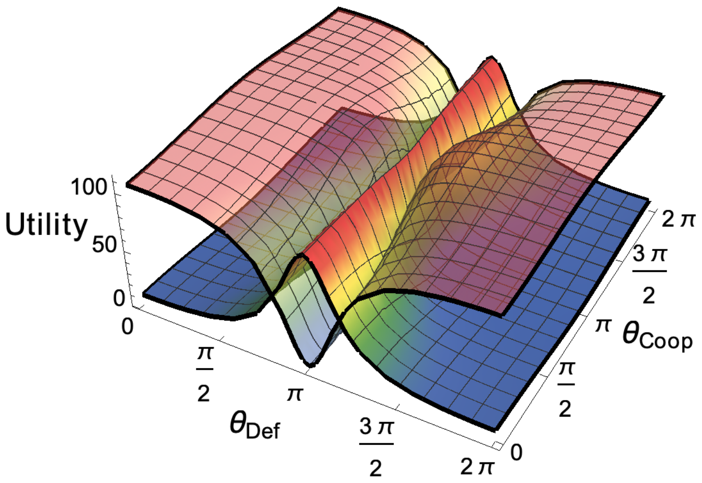



```
In[55]:= (* ####### EXPECTED UTILITY OF COOPERATION AND DEFECT ACCORDING TO THE QUANTUM-
  LIKE INFLUENCE DIAGRAMS ######## *)
```

In[56]:=
```
(* One can find that there are points in the graph that
 favour a cooperative action rather than a defect action *)
```

In[51]:=
```
Show[utilCoop, utilDef, PlotRange → All]
```

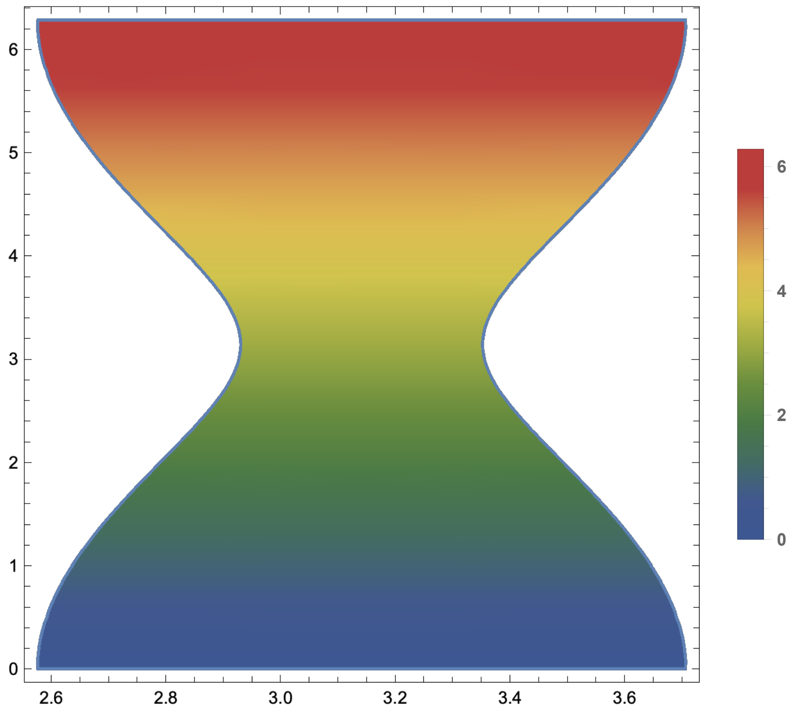
Out[51]=



In[62]:=
```
(* ################################## REGION PLOTS
 #################################################### *)
(* One can see in the above Figure that with quantum interference,
one can influence the players' utilities in
 such a way that it favours a cooperative action. *)
```

In[58]:=
```
RegionPlot[UtilityX2c ≥  UtilityX2d, {θDef, 0, 2 π},
 {θCoop, 0, 2 π}, AxesLabel → Automatic, PlotRange → All,
 ColorFunction → "DarkRainbow", PlotLegends → Automatic]
```

Out[58]=

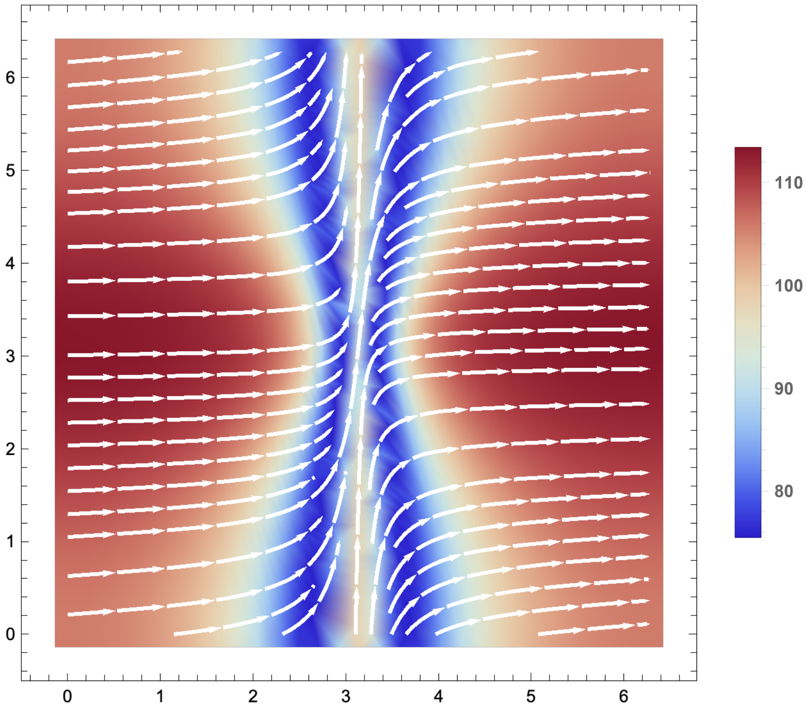


In[63]:=
```
(* ############## ANALYSE EVOLUTION OF INTERFERENCE EFFECTS IN QUANTUM-
 LIKE EXPECTED UTILITIES ################### *)
```

In[64]:=
```
(* Evolution of Decision-
 Makers beliefs to Defect towards beliefs to Cooperate *)
```

In[61]:=
```
StreamDensityPlot[{UtilityX2d, UtilityX2c}, {θDef, 0, 2 π}, {θCoop, 0, 2 π},
 ColorFunction → "ThermometerColors", PlotLegends → Automatic,
 AxesLabel → Automatic, StreamStyle → {White, Thick}]
```

Out[61]=




```
(* Evolution of Decision-
 Makers beliefs to Cooperate towards beliefs to Defect *)
```

In[65]:=
```
StreamDensityPlot[{UtilityX2c, UtilityX2d}, {θDef, 0, 2 π}, {θCoop, 0, 2 π},
 ColorFunction → "ThermometerColors", PlotLegends → Automatic,
 AxesLabel → Automatic, StreamStyle → {White, Thick}]
```

Out[65]=