

Physik der sozio-ökonomischen Systeme *mit dem Computer*

*JOHANN WOLFGANG GOETHE UNIVERSITÄT
10.05.2022*

MATTHIAS HANAUSKE

*FRANKFURT INSTITUTE FOR ADVANCED STUDIES
JOHANN WOLFGANG GOETHE UNIVERSITÄT
INSTITUT FÜR THEORETISCHE PHYSIK
ARBEITSGRUPPE RELATIVISTISCHE ASTROPHYSIK
D-60438 FRANKFURT AM MAIN
GERMANY*

4. Vorlesung

Plan für die heutige Vorlesung

- Kurze Wiederholung der Inhalte der 3. Vorlesung
- Einführung in die Evolutionäre Spieltheorie
 - Die Differentialgleichung eines evolutionären, symmetrischen (2x2)-Spiels
 - Dominante Spiele
 - Koordinationsspiele
 - Anti-Koordinationsspiele
 - Das System von Differentialgleichung eines evolutionären, unsymmetrischen (2x2)-Spiels (Bi-Matrix Spiele)
 - Eckenspiele (Corner Class Games)
 - Sattelpunktspiele (Saddle Class Games)
 - Zentrumsspiele (Center Class Games)

Inhalte Vorlesung 3

- Symmetrische und nicht symmetrische Spiele
- Klassen symmetrischer (2x2)-Spiele
- Beispiel eines asymmetrischen Spiels: Kampf der Geschlechter
- Einführung in die Evolutionäre Spieltheorie
 - Die Replikatorodynamik der Quasispezies
 - Die Differentialgleichung eines evolutionären (2xm)-Spiels
 - Die Differentialgleichung eines evolutionären (2x2)-Spiels (Bi-Matrix Spiel)
 - Die Differentialgleichung eines evolutionären, symmetrischen (2x2)-Spiels
 - Definition: Evolutionär stabile Strategien (ESS)

Klassen symmetrischer (2x2)-Spiele

	Spieler B Strategie 1 $y=1$	Spieler B Strategie 1 $y=0$
Spieler A Strategie 1 $x=1$	(a, a)	(b, c)
Spieler A Strategie 2 $x=0$	(c, b)	(d, d)

Symmetrisches
(2x2)-Spiel

Symmetrische (2x2)-Spiele lassen sich in drei unterschiedliche Klassen gliedern:

1. Dominante Spiele
2. Koordinationsspiele
3. Anti-Koordinationsspiele

Die Klasse der dominanten Spiele ($a > c$ und $b > d$ bzw. $a < c$ und $b < d$)

Bei dieser Spielklasse dominiert eine Strategie die andere. Es existiert nur ein reines Nash-Gleichgewicht welches die dominante Strategie des Spiels darstellt. Dieser Fall tritt ein, falls:

$a > c$ und $b > d$: Strategie 1 dominiert Strategie 2; dominante Strategie bei $(x,y)=(1,1)$.

$a < c$ und $b < d$: Strategie 2 dominiert Strategie 1; dominante Strategie bei $(x,y)=(0,0)$.

Koordinationsspiele ($a > c$ und $b < d$)

Ein Koordinationsspiel existiert, falls die Parameter a , b , c und d der Auszahlungsmatrix die folgenden Bedingungen erfüllen: $a > c$ und $b < d$. Bei dieser Spielklasse existieren drei Nash-Gleichgewichte, ein gemischtes Nash-Gleichgewicht und zwei reine, symmetrische Nash-Gleichgewicht bei $(x,y)=(0,0)$ und $(x,y)=(1,1)$.

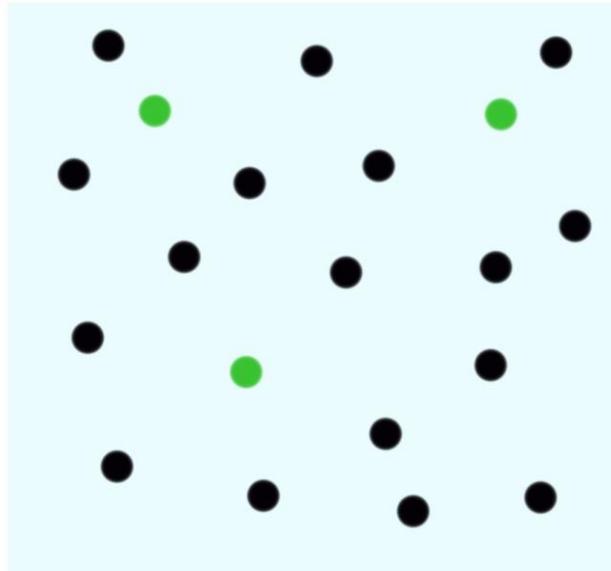
Anti-Koordinationsspiele ($a < c$ und $b > d$)

Ein Anti-Koordinationsspiel existiert, falls die Parameter a , b , c und d der Auszahlungsmatrix die folgenden Bedingungen erfüllen: $a < c$ und $b > d$. Bei dieser Spielklasse existieren drei Nash-Gleichgewichte, ein gemischtes Nash-Gleichgewicht und zwei reine, unsymmetrische Nash-Gleichgewicht bei $(x,y)=(0,1)$ und $(x,y)=(1,0)$.

Evolutionäre Spieltheorie

Symmetrische (2x2)-Spiele einer Population

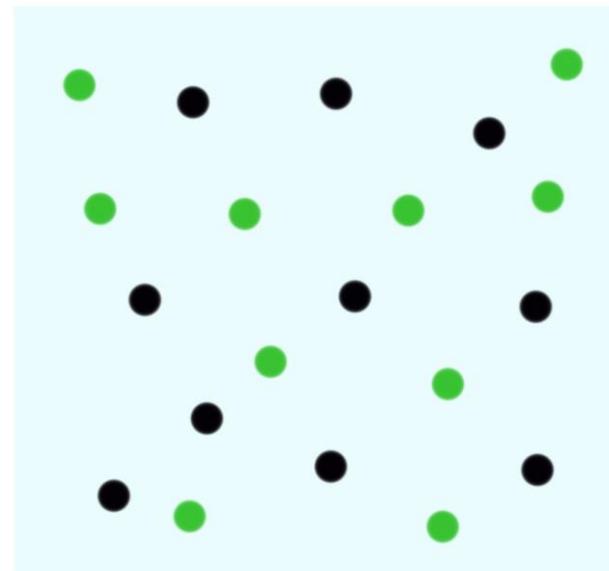
Die evolutionäre Spieltheorie betrachtet die zeitliche Entwicklung des strategischen Verhaltens einer gesamten Spielerpopulation.



$$x(0)=0.15$$



zeitliche
Entwicklung
der
Population



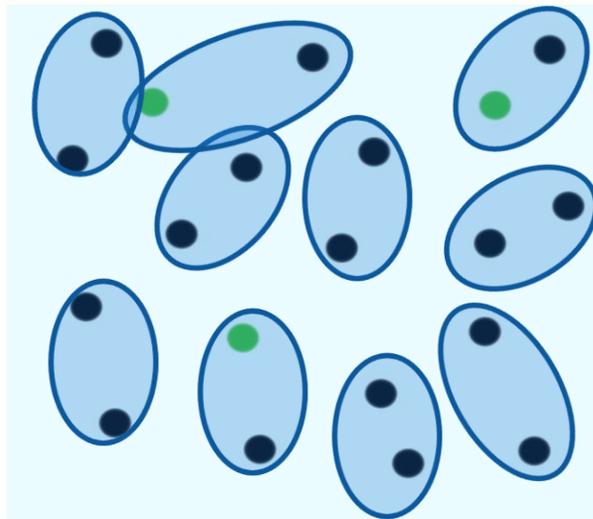
$$x(10)=0.5$$

Mögliche Strategien: (grün, schwarz), Parameter t stellt die „Zeit“ dar.
 $x(t)$: Anteil der Spieler, die im Zeitpunkt t die Strategie „grün“ spielen.

Evolutionäre Spieltheorie

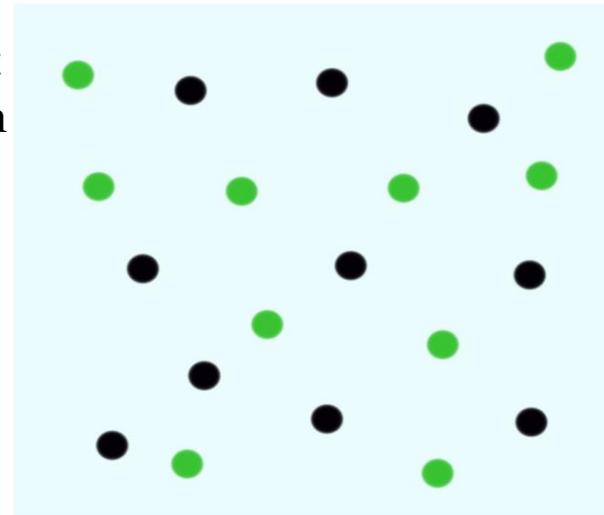
Symmetrische (2x2)-Spiele einer Population

Die einzelnen Akteure innerhalb der betrachteten Population spielen ein andauernd sich wiederholendes Spiel miteinander, wobei sich jeweils zwei Spieler zufällig treffen, das Spiel spielen und danach zu dem nächsten Spielpartner wechseln.



$$x(0)=0.15$$

Die Anfangspopulation von Spielern spielt zum Zeitpunkt $t=0$ das erste Mal das Spiel. Die Spieler wählen im Mittel zu 15% die grüne Strategie.



$$x(10)=0.5$$

Das evolutionäre Spiel schreitet voran und die grüne Strategie wird für die Spieler zunehmend attraktiver. Zum Zeitpunkt $t=10$ spielen schon 50% grün.

$$\begin{aligned}
\frac{dx_i^A(t)}{dt} &= \left[\underbrace{\sum_{l=1}^{m_B} \$_{il}^A x_l^B(t)}_{\text{Fitness der Strategie i}} - \underbrace{\sum_{l=1}^{m_B} \sum_{k=1}^{m_A} \$_{kl}^A x_k^A(t) x_l^B(t)}_{\text{Durchschn. Fitness der Population A}} \right] x_i^A(t) \quad (1) \\
\frac{dx_j^B(t)}{dt} &= \left[\underbrace{\sum_{l=1}^{m_A} \$_{lj}^B x_l^A(t)}_{\text{Fitness der Strategie j}} - \underbrace{\sum_{l=1}^{m_A} \sum_{k=1}^{m_B} \$_{lk}^B x_l^A(t) x_k^B(t)}_{\text{Durchschn. Fitness der Population B}} \right] x_j^B(t) \quad ,
\end{aligned}$$

wobei $x_i^A(t)$, $i = 1, 2, \dots, m_A$ und $x_j^B(t)$, $j = 1, 2, \dots, m_B$ die Anteile der in den Spielergruppen A und B zur Zeit t gewählten Strategien widerspiegeln und in der Soziobiologie den Frequenzen der *Quasispezies* entsprechen.

Nimmt man zusätzlich ein symmetrisches Spiel an ($\hat{\$} := \hat{\$}^A = \left(\hat{\$}^B\right)^T$), in welchem die Auszahlungswerte (Fitness-Werte) der Populationsgruppen gleich sind, so kann man die beiden Gruppen von ihrer mathematischen Struktur her als ununterscheidbare Spielergruppen mit identischen Populationsvektoren $x(t) = y(t)$ annehmen. Die Differentialgleichung schreibt sich dann wie folgt:

$$\frac{dx(t)}{dt} = [(\$_{11} - \$_{21})(x - x^2) + (\$_{12} - \$_{22})(1 - 2x + x^2)] x(t) =: g(x) \quad (3)$$

Verallgemeinert man diese Differentialgleichung wieder auf mehr als zwei Strategien, so kann man abkürzend die folgende Formulierung schreiben:

$$\frac{d\vec{x}}{dt} = \hat{\mathbf{x}} \left(\hat{\$} \vec{x} \right) - \left(\left(\hat{\$} \vec{x} \right)^T \vec{x} \right) \vec{x}$$

Replikatordynamik

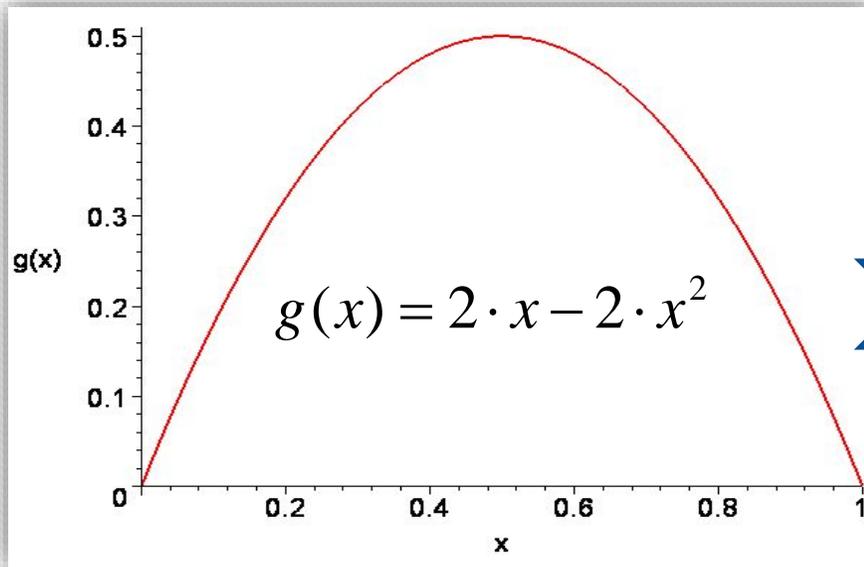
(für das Gefangenendilemma)

Die Differentialgleichung der Replikatordynamik für das Gefangenendilemma lautet:

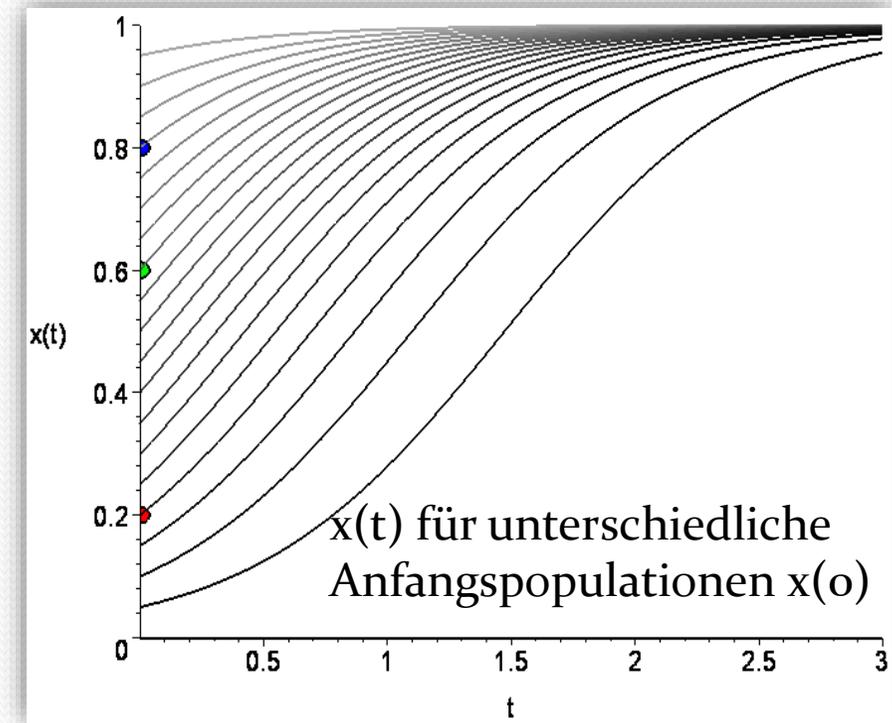
$$\dot{x}(t) := \frac{dx(t)}{dt} = 2 \cdot x(t) - 2 \cdot (x(t))^2 = g(x(t))$$

$$g(x(t)) := x(t) \cdot \left((-7 - (-9)) \cdot (x(t) - x(t)^2) + (-3 - (-1)) \cdot (2 \cdot x(t) - 1 - x(t)^2) \right)$$

	Ge	Sc
Ge	(-7, -7)	(-1, -9)
Sc	(-9, -1)	(-3, -3)



Beispiel: Gefangenendilemma
 $g(x) = g(x(t))$ im Bereich $[0,1]$ dargestellt



Evolutionär stabile Strategien (ESS)

- Evolutionär stabile Strategien sind die stabilen Endzustände der Häufigkeitsverteilung $x(t)$:

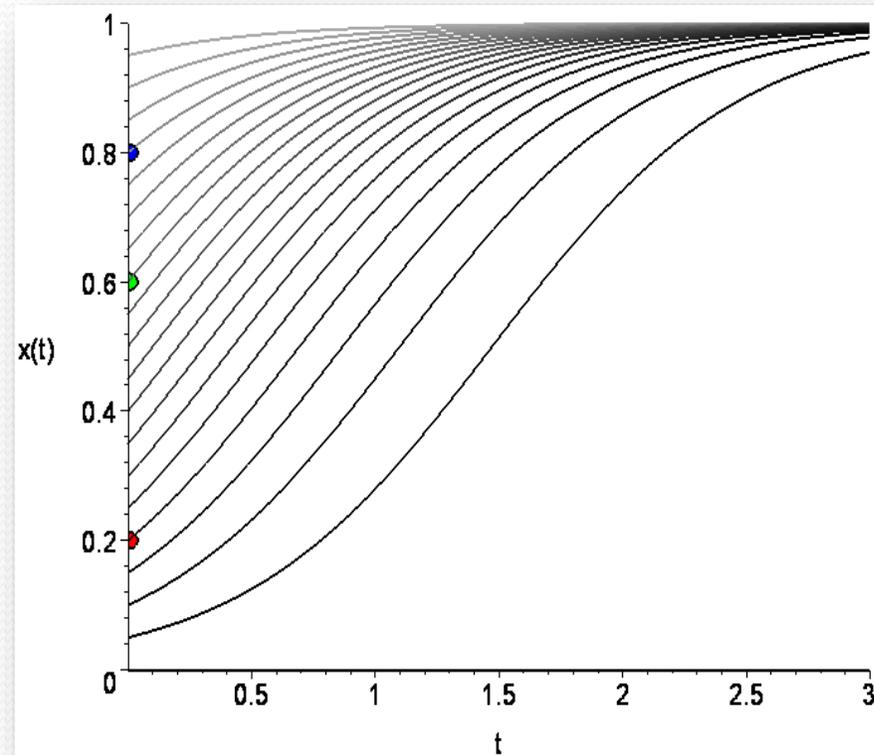
$$\lim_{t \rightarrow \infty} (x(t))$$

Eine notwendige (aber nicht hinreichende) Bedingung für die Existenz einer ESS ist, dass diese ein Nash – Gleichgewicht des zugrundeliegenden Spiels ist.

Beispiel:

Gefangenendilemma ähnliche Spiele
Für die ESS des evolutionären Gefangenendilemma – Spiels ergibt sich:

$$\lim_{t \rightarrow \infty} (x(t)) = 1 \Rightarrow \text{alle "gestehen"}$$



Replikatorodynamik

(für das Hirschjagt-Spiel)

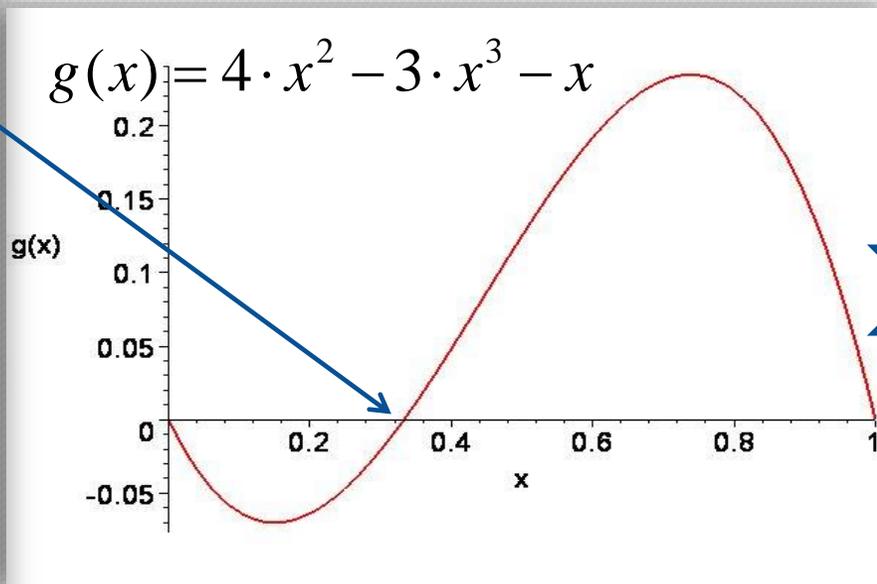
Die Differentialgleichung der Replikatorodynamik für das Hirschjagt-Spiel lautet:

$$\dot{x}(t) := \frac{dx(t)}{dt} = 4 \cdot (x(t))^2 - 3 \cdot (x(t))^3 - x(t) =: g(x(t))$$

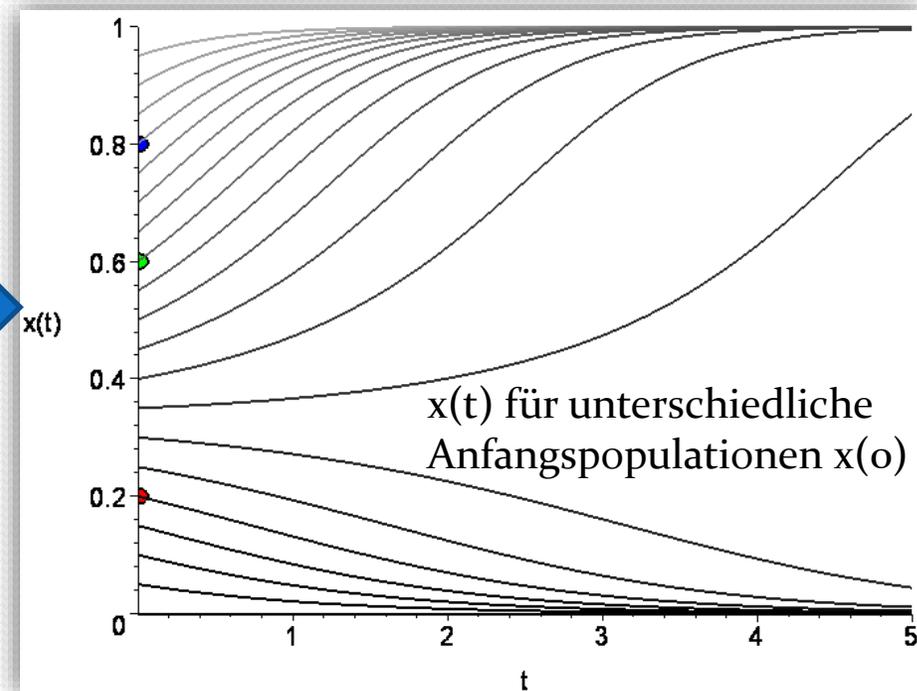
$$g(x(t)) := x(t) \cdot \left((2 - 0) \cdot (x(t) - x(t)^2) + (5 - 4) \cdot (2 \cdot x(t) - 1 - x(t)^2) \right)$$

	Hasen	Hirsch
Hasen	(2, 2)	(4, 0)
Hirsch	(0, 4)	(5, 5)

Die interne Nullstelle der Funktion $g(x)$ entspricht dem Wert des gemischten Nash-Gleichgewichts und bestimmt den separations-Anfangswert der Population



Beispiel: Hirschjagt-Spiel
 $g(x)=g(x(t))$ im Bereich $[0,1]$ dargestellt



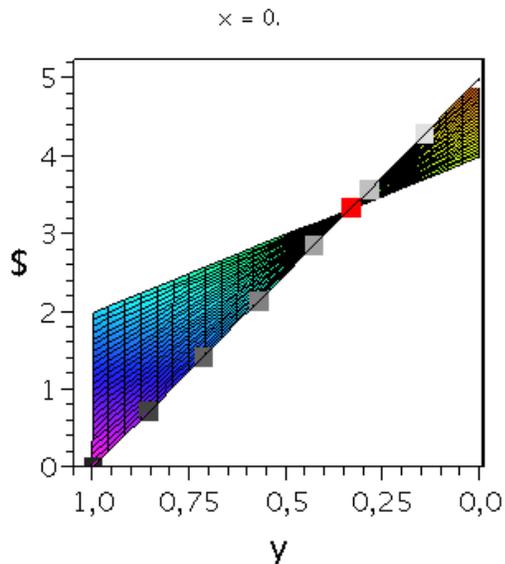
Das Nash-Gleichgewicht in gemischten Strategien

Ein Spezialfall des Nash-Gleichgewichts besteht, falls die partielle Ableitung der gemischten Auszahlungsfläche verschwindet. Man nennt dann ein solches Nash-Gleichgewicht $(\tilde{s}^{A*}, \tilde{s}^{B*})$ ein internes Nash-Gleichgewicht bzw. ein Nash-Gleichgewicht in gemischten Strategien.

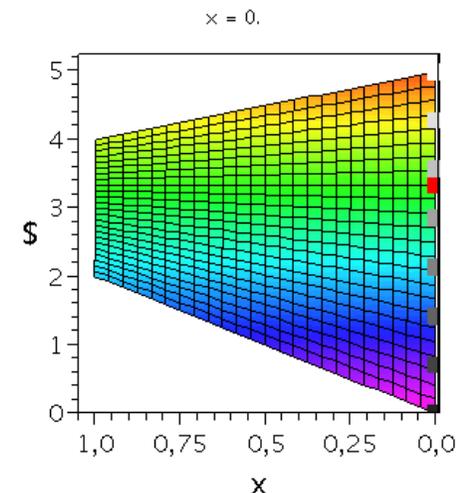
Nash-Gleichgewicht in gemischten Strategien:

$$\left. \frac{\partial \tilde{\$}^A(\tilde{s}^A, \tilde{s}^B)}{\partial \tilde{s}^A} \right|_{\tilde{s}^B = \tilde{s}^{B*}} = 0 \quad \forall \tilde{s}^A \in [0, 1] , \tilde{s}^{B*} \in]0, 1[$$

$$\left. \frac{\partial \tilde{\$}^B(\tilde{s}^A, \tilde{s}^B)}{\partial \tilde{s}^B} \right|_{\tilde{s}^A = \tilde{s}^{A*}} = 0 \quad \forall \tilde{s}^B \in [0, 1] , \tilde{s}^{A*} \in]0, 1[$$



Das gemischte Nash-Gleichgewicht im Hirschjagd-Spiel liegt bei der Strategien-kombination $(x=1/3, y=1/3)$. Das gemischte Nash-Gleichgewicht entspricht der internen Nullstelle der Funktion $g(x)$.



Replikatordynamik

(für das Angsthasen-Spiel)

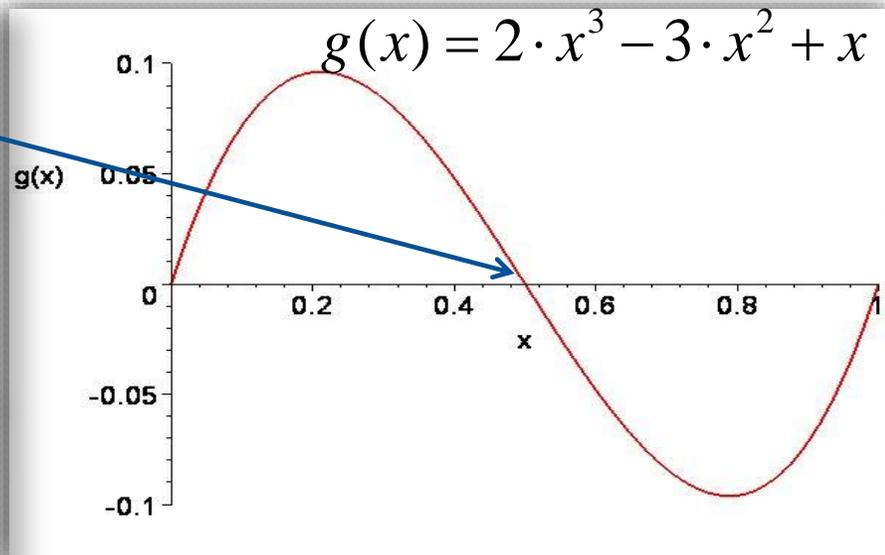
Die Differentialgleichung der Replikatordynamik für das Angsthasen-Spiel lautet:

$$\dot{x}(t) := \frac{dx(t)}{dt} = 2 \cdot (x(t))^3 - 3 \cdot (x(t))^2 + x(t) = g(x(t))$$

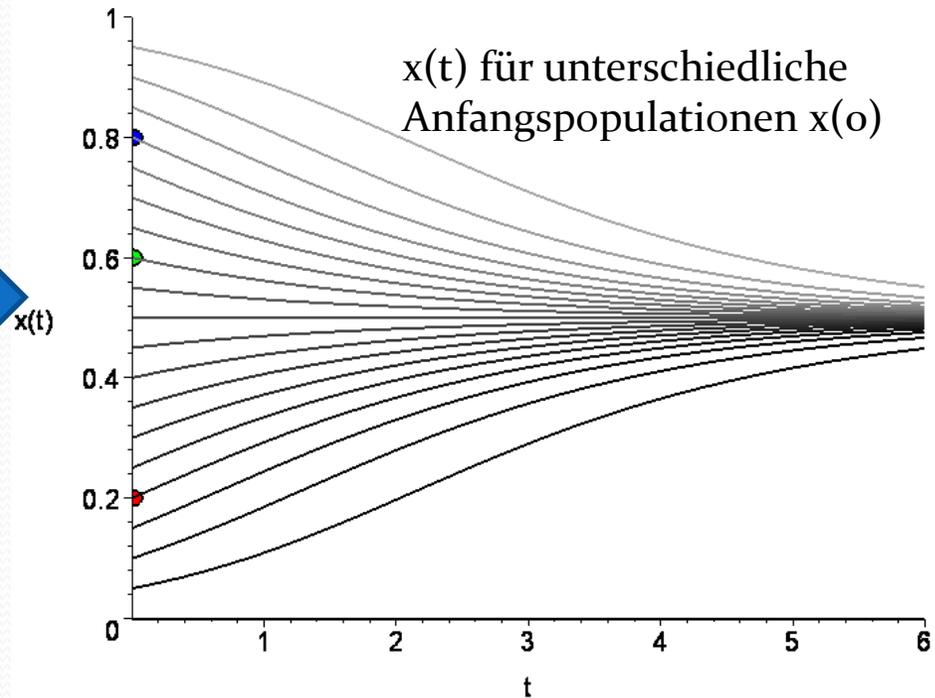
$$g(x(t)) := x(t) \cdot \left((-1 - 0) \cdot (x(t) - x(t)^2) + (1 - 2) \cdot (2 \cdot x(t) - 1 - x(t)^2) \right)$$

	Springe nicht	Springe
Springe nicht	(-1, -1)	(2, 0)
Springe	(0, 2)	(1, 1)

Die interne Nullstelle der Funktion $g(x)$ entspricht dem Wert des gemischten Nash-Gleichgewichts und bestimmt die Lage der ESS



Beispiel: Angsthasen-Spiel
 $g(x) = g(x(t))$ im Bereich $[0,1]$ dargestellt



Evolutionär Stabile Strategien

- Betrachten Sie die folgenden Beispiele:

Beispiel 1:			Beispiel 2:			Beispiel 3:		
	Kugel	Keine Kugel		Kugel	Keine Kugel		Kugel	Keine Kugel
Kugel	(0, 0)	(2, -1)	Kugel	(-1, -1)	(3, 0)	Kugel	(0, 0)	(2, -2)
Keine Kugel	(-1, 2)	(1, 1)	Keine Kugel	(0, 3)	(1, 1)	Keine Kugel	(-2, 2)	(3, 3)

1. Geben Sie mögliche dominante Strategien und Nash-Gleichgewichte der Spiele an.
2. Bestimmen und zeichnen Sie die Funktion $g(x)$ für alle drei Spiele?
3. Berechnen Sie die Nullstellen der Funktion $g(x)$ ($g(x)=0$).
4. Geben Sie die evolutionär stabilen Strategien der Spiele an?

Dominante Strategien und Nash-Gleichgewichte

Beispiel 1

	Kugel	Keine Kugel
Kugel	(0, 0)	(2, -1)
Keine Kugel	(-1, 2)	(1, 1)



Das erste Spiel besitzt nur ein Nash-Gleichgewicht das gleichzeitig die dominante Strategie des Spiels ist (Kugel, Kugel). Das Spiel gehört zur Klasse der dominanten Spiele.

Beispiel 2

	Kugel	Keine Kugel
Kugel	(-1, -1)	(3, 0)
Keine Kugel	(0, 3)	(1, 1)



Das zweite Spiel besitzt keine dominante Strategie, aber zwei unsymmetrische Nash-Gleichgewicht in reinen Strategien ((K, KK) und (KK, K)) und ein gemischtes Nash-Gleichgewicht (0.67 K, 0.33 KK). Das Spiel gehört zur Klasse der Anti-Koordinationsspiele.

Beispiel 3

	Kugel	Keine Kugel
Kugel	(0, 0)	(2, -2)
Keine Kugel	(-2, 2)	(3, 3)



Das dritte Spiel besitzt ebenfalls keine dominante Strategie, aber zwei symmetrische Nash-Gleichgewicht in reinen Strategien ((K, K) und (KK, KK)) und ein gemischtes Nash-Gleichgewicht (0.33 K, 0.67 KK). Das Spiel gehört zur Klasse der Koordinationsspiele.

Nullstellen der Funktion $g(x)$

Beispiel 1:

$$g(x) = x - x^2$$

$$x - x^2 = 0$$

$$x \cdot (1 - x) = 0 \Rightarrow x_1 = 0$$

$$1 - x = 0 \Rightarrow x_2 = 1$$

Die Funktion hat zwei Nullstellen:

$$x_1 = 0 \text{ und } x_2 = 1$$

Beispiel 3:

Die Funktion hat drei Nullstellen:

$$g(x) = -3 \cdot x^3 + 4 \cdot x^2 - x$$

$$x_1 = 0, x_2 = 1 \text{ und } x_3 = \frac{1}{3}$$

Beispiel 2:

$$g(x) = 3 \cdot x^3 - 5 \cdot x^2 + 2 \cdot x$$

$$x \cdot (3 \cdot x^2 - 5 \cdot x + 2) = 0 \Rightarrow x_1 = 0$$

$$3 \cdot x^2 - 5 \cdot x + 2 = 0 \quad | /3$$

$$x^2 - \frac{5}{3} \cdot x + \frac{2}{3} = 0 \quad | \text{ p-q Formel}$$

$$x_{2/3} = \frac{5}{6} \pm \sqrt{\left(\frac{5}{6}\right)^2 - \frac{2}{3}}$$

$$x_{2/3} = \frac{5}{6} \pm \sqrt{\frac{25 - 24}{36}}$$

$$x_{2/3} = \frac{5}{6} \pm \frac{1}{6} \Rightarrow x_2 = 1, x_3 = \frac{2}{3}$$

Die Funktion hat drei Nullstellen:

$$x_1 = 0, x_2 = 1 \text{ und } x_3 = \frac{2}{3}$$

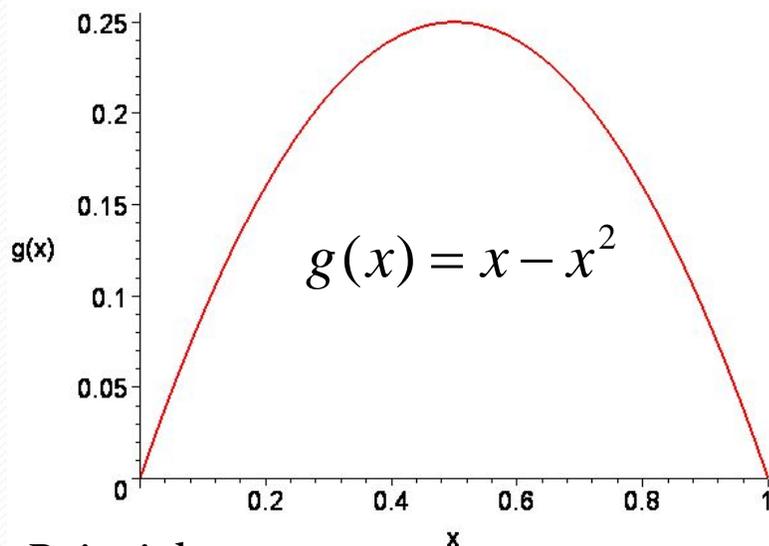
Evolutionäre Strategien (Beispiel 1)

	Kugel	Keine Kugel
Kugel	(0, 0)	(2, -1)
Keine Kugel	(-1, 2)	(1, 1)

Die Differentialgleichung der Replikatorodynamik für das erste Beispiel lautet:

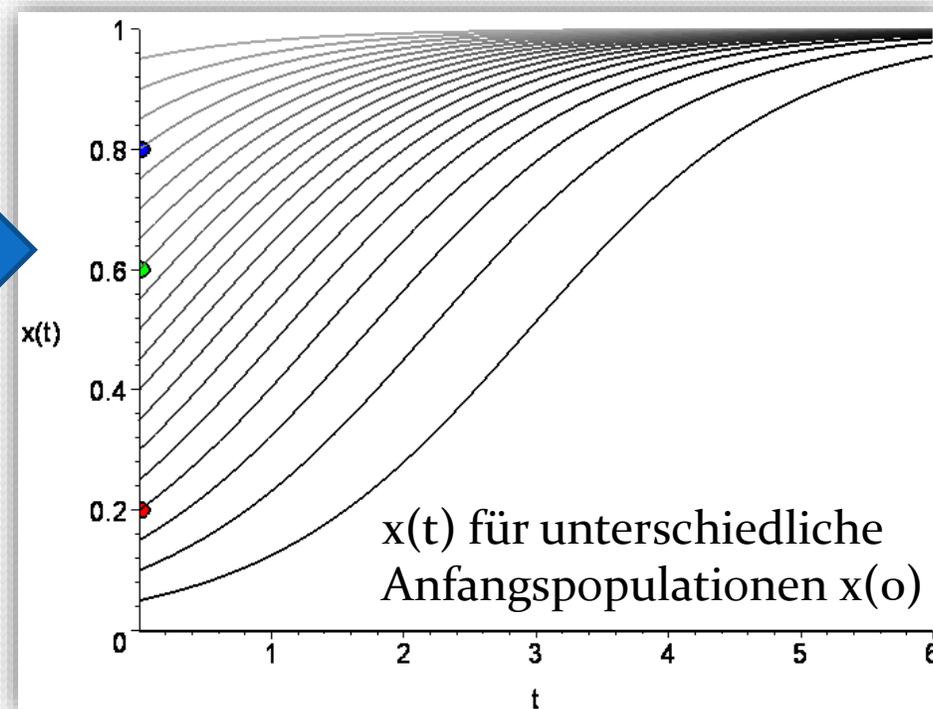
$$\frac{dx(t)}{dt} = g(x(t)) = x(t) - (x(t))^2$$

Eine ESS bei $x=1$



Beispiel 1:
 $g(x)=g(x(t))$ im Bereich $[0,1]$ dargestellt

Da es sich bei diesem Beispiel um ein dominantes, symmetrisches (2x2)-Spiel handelt und die Funktion $g(x)$ im relevanten Bereich ($x=[0,1]$) immer größer-gleich Null ist, strebt der Populationsanteil der Kugel-Spieler unabhängig vom Anfangswert immer gegen 1.



Evolutionäre Strategien (Beispiel 2)

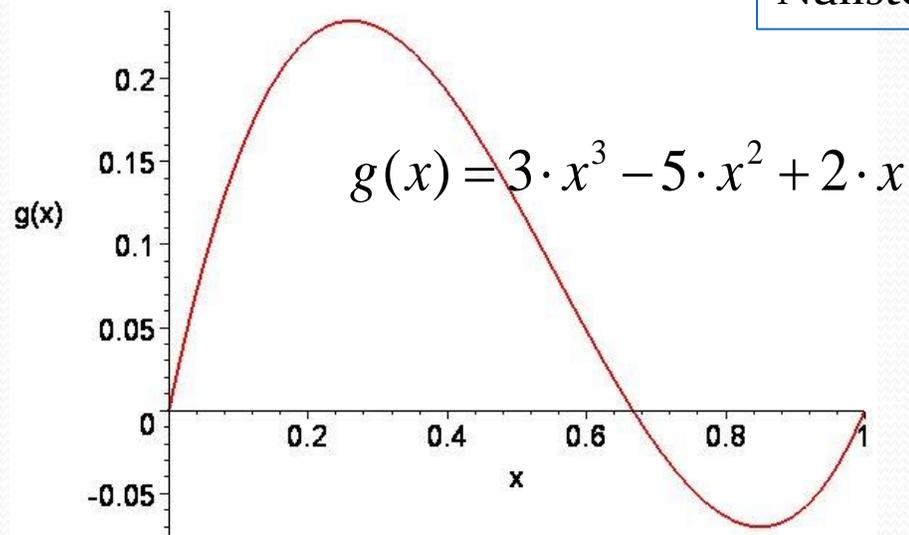
	Kugel	Keine Kugel
Kugel	$(-1, -1)$	$(3, 0)$
Keine Kugel	$(0, 3)$	$(1, 1)$

Die Differentialgleichung der Replikatorodynamik für das zweite Beispiel lautet:

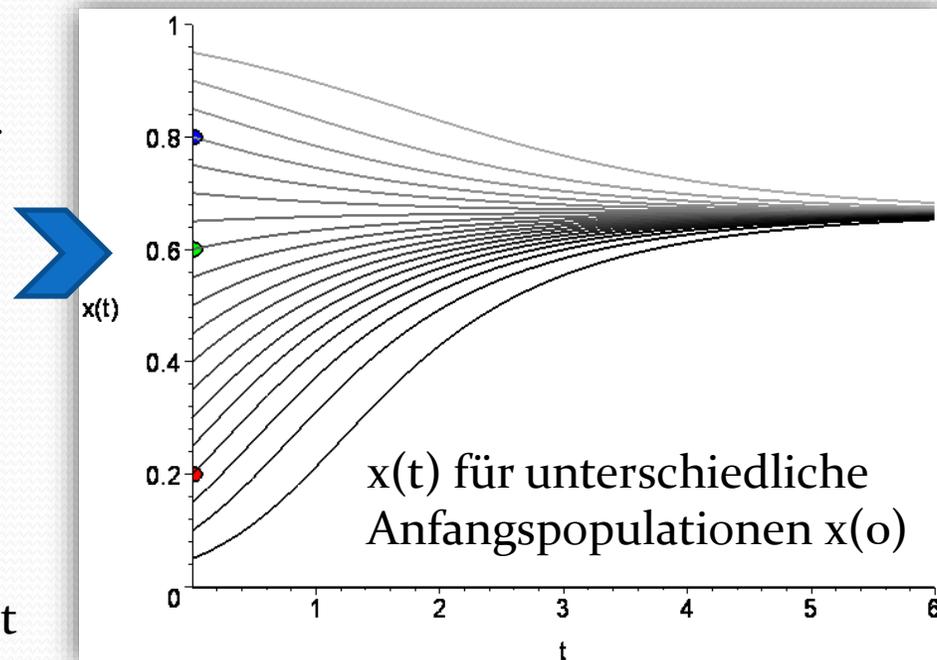
$$\frac{dx(t)}{dt} = g(x(t)) = 3 \cdot (x(t))^3 - 5 \cdot (x(t))^2 + 2 \cdot x(t)$$

Eine ESS bei $x=0.67$

Da es sich bei diesem Beispiel um ein symmetrisches Anti-Koordinationsspiel handelt, strebt der Populationsanteil der Kugel-Spieler unabhängig vom Anfangswert immer zu dem gemischten Nash-Gleichgewicht, was identisch mit der mittleren Nullstelle der Funktion $g(x)$ ist ($x=0.67$).



Beispiel 2:
 $g(x)=g(x(t))$ im Bereich $[0,1]$ dargestellt



Evolutionäre Strategien (Beispiel 3)

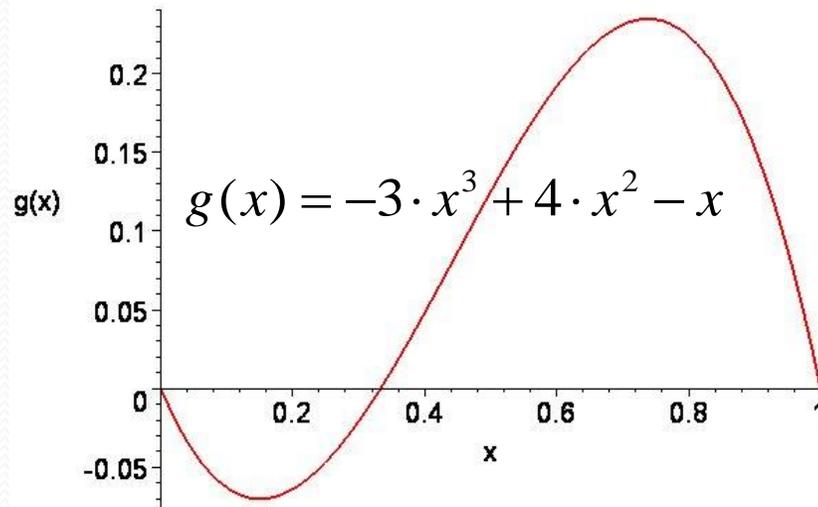
	Kugel	Keine Kugel
Kugel	(0, 0)	(2, -2)
Keine Kugel	(-2, 2)	(3, 3)

Die Differentialgleichung der Replikatorodynamik für das zweite Beispiel lautet:

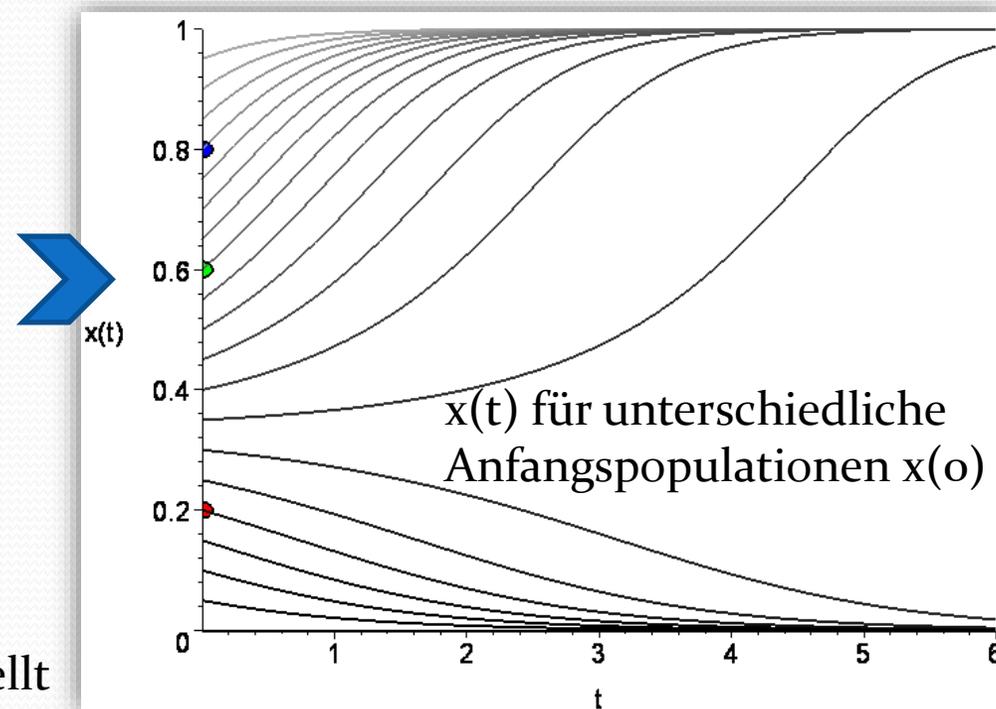
$$\frac{dx(t)}{dt} = g(x(t)) = -3 \cdot (x(t))^3 + 4 \cdot (x(t))^2 - x(t)$$

Zwei ESSs : ($x=1$ und $x=0$)

Da es sich bei diesem Beispiel um ein symmetrisches Koordinationsspiel handelt, strebt der Populationsanteil der Kugel-Spieler abhängig vom Anfangswert zu einem der beiden reinen Nash-Gleichgewichte ($x=1$ oder $x=0$).

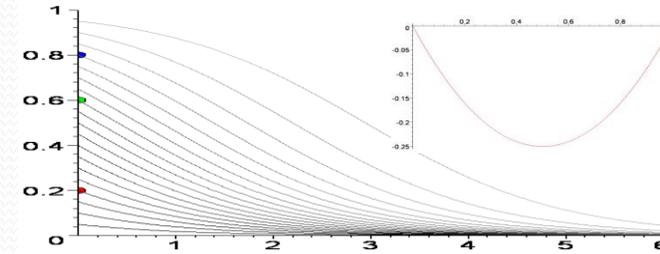


Beispiel 3:
 $g(x)=g(x(t))$ im Bereich $[0,1]$ dargestellt

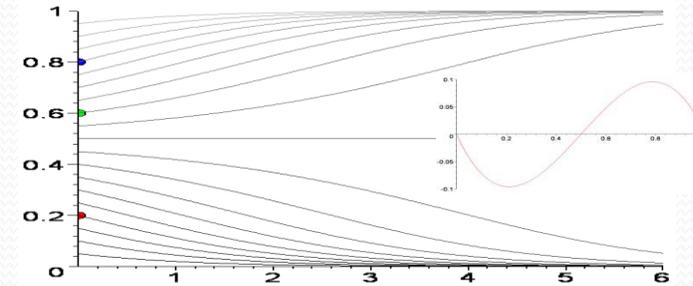


Klassifizierung von evolutionären, symmetrischen (2x2)-Spielen

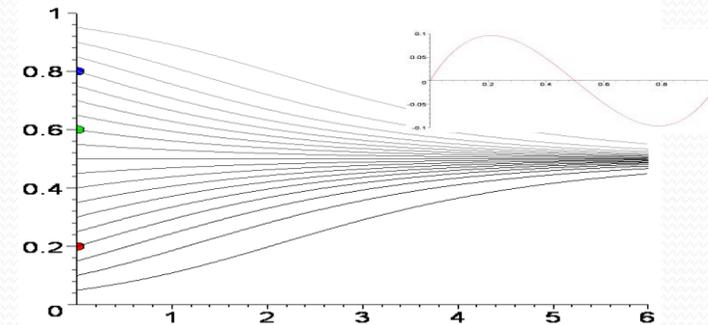
- **Dominante Spiele**
(2. Strategie dominiert 1.Strategie)
Es existiert ein Nash - Gleichgewicht, welches die anderen Strategien dominiert. ESS bei $x=0$.



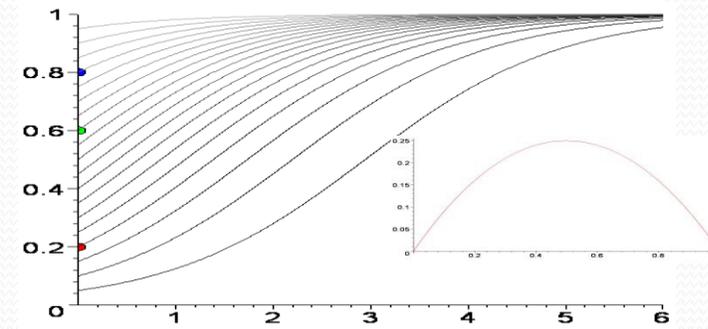
- **Koordinationsspiele**
Es existieren drei Nash - Gleichgewichte und zwei reine ESS, die abhängig von der Anfangsbedingung realisiert werden.



- **Anti - Koordinationsspiele**
Es existieren drei Nash - Gleichgewichte aber nur eine gemischte ESS, die unabhängig von der Anfangsbedingung realisiert wird.



- **Dominante Spiele**
(1. Strategie dominiert 2.Strategie)
Es existiert ein Nash - Gleichgewicht, welches die anderen Strategien dominiert. ESS bei $x=1$.



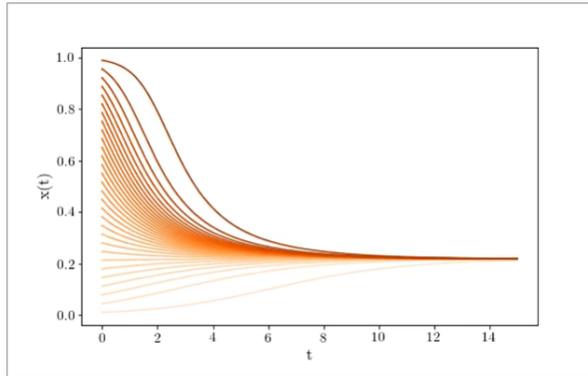
Internetseite der Vorlesung 4

Auf der Internetseite der Vorlesung 4, unter [Weiterführende Links](#), sind ein [Jupyter Notebook](#) und zwei [Python Programme](#) zum Herunterladen bereitgestellt, welche die numerische Lösung von symmetrischen evolutionären 2-Personen 2-Strategien Spielen behandeln.

Vorlesung 4

Die Lösungen der im rechten Panel dargestellten Differentialgleichungen der *evolutionäre Spieltheorie* werden in dieser Vorlesung numerisch gelöst und die möglichen zeitlichen Entwicklungen der Populationen in Klassen untergliedert. Die entsprechenden Python Programme (Python Skripts) und Jupyter Notebooks finden Sie weiter unten unter [Weiterführende Links](#).

Evolutionäre symmetrische (2 × 2) Spiele



Die im unteren Bereich des rechten Panels dieser Vorlesung dargestellte Gleichung der Reproduktionsdynamik lässt sich nach Spezifikation der symmetrischen Auszahlungsmatrix $\$$ auf unterschiedlichste Populationsspiele anwenden. Generell lassen sich symmetrische (2 × 2) Spiele und somit auch die Lösungen der Differentialgleichung der Reproduktionsdynamik für eine Populationsgruppe mit zwei Quasispezies in drei unterschiedliche Spielklassen gliedern: Dominante Spiele, Koordinationsspiele und Anti-Koordinationsspiele. Bei dominanten Spielen entwickelt sich, unabhängig von dem gewählten Anfangszustand, die Entscheidungswahl der Population stets zu einem Zustand, indem alle Spieler die dominante Strategie spielen - die dominante Strategie entspricht somit der evolutionär stabilen Strategie (ESS). Bei Koordinationsspielen entwickelt sich die Population, abhängig von ihrem Anfangszustand, entweder zu

einem Zustand in dem alle Spieler die Strategie 2 ($x=0$) oder alle die Strategie 1 ($x=1$) wählen - man sagt es gibt zwei ESSs in Koordinationsspielen. Der kritische Anfangswert, der die beiden ESSs voneinander separiert, entspricht dem gemischten Nash-Gleichgewicht des zugrundeliegenden Koordinationsspiels und ist außerdem die interne Nullstelle der Funktion $g(x)$. Bei Anti-Koordinationsspielen entwickelt sich die Population, unabhängig von ihrem Anfangszustand, stets zu einem Zustand der Koexistenz beider Strategien, wobei die für große Zeiten asymptotisch erreichte Populationszusammensetzung durch den Wert des internen gemischten Nash-Gleichgewichtes gegeben ist und die einzige evolutionär stabile Strategie darstellt. Die nebenstehende Animation zeigt die zeitliche Entwicklung der Population im [Falke-Taube Spiel](#) für unterschiedliche Anfangszusammensetzung der Population, wobei der Parameter V (Grad der negativen Auswirkung, wenn zwei Falke-Strategien aufeinander treffen) variiert wurde. Variiert man den Parameter V (hier $V=[1,5]$), so verschiebt man das gemischte Nash-Gleichgewicht und somit die ESS (siehe [Klassifizierung evolutionärer symmetrischer \(2x2\)-Spiele](#)).

Evolutionäre unsymmetrische (2 × 2) Spiele (Bi-Matrix Spiele)



Die im oberen Bereich des rechten Panels dieser Vorlesung dargestellten Gleichungen der Reproduktionsdynamik lassen

Vorlesung 4

In der vorigen Vorlesung wurden die zugrundeliegenden Differentialgleichungen der *evolutionäre Spieltheorie* für zeitlich sich wiederholende, unsymmetrische (2 Personen)-(m Strategien) Spiele dargestellt. Wir beschränken uns im Folgenden auf den 2-Strategien Fall ($m_A = m_B = 2$), lassen jedoch weiter eine Unsymmetrie der Auszahlungsmatrix zu (evolutionäre Bi-Matrix Spiele). Die beiden Komponenten der zweidimensionalen gruppenspezifischen Populationsvektoren lassen sich dann aufgrund ihrer Normalisierungsbedingung, auf eine Komponente reduzieren ($x_2^A = 1 - x_1^A$ und $x_2^B = 1 - x_1^B$). Das zeitliche Verhalten der Komponenten der Populationsvektoren (Gruppe A: $x(t) := x_1^A(t)$ und Gruppe B: $y(t) := x_1^B(t)$) wird in der Reproduktionsdynamik mittels des folgenden Systems von Differentialgleichungen beschrieben:

$$\begin{aligned} \frac{dx(t)}{dt} &= g_A(x, y) \quad , \quad \frac{dy(t)}{dt} = g_B(x, y) \\ g_A(x, y) &:= [(\$_{11}^A + \$_{22}^A - \$_{12}^A - \$_{21}^A) y(t) + (\$_{12}^A - \$_{22}^A)] (x(t) - (x(t))^2) \\ g_B(x, y) &:= [(\$_{11}^B + \$_{22}^B - \$_{12}^B - \$_{21}^B) x(t) + (\$_{12}^B - \$_{22}^B)] (y(t) - (y(t))^2) \end{aligned}$$

In dieser Vorlesung betrachten wir jedoch zunächst die Lösungen der Differentialgleichung für sich wiederholende, symmetrische (2 Personen)-(2 Strategien) Spiele. Aufgrund der Symmetrie des Spiels besteht die betrachtete Population nun nur aus einer Gruppe von ununterscheidbaren Spielern (beschrieben durch den Populationsvektor $x(t)$). Die Replikatordynamik der evolutionäre Spieltheorie vereinfacht sich in diesem Spezialfall auf nur eine nichtlineare Differentialgleichung erster Ordnung:

$$\begin{aligned} \frac{dx(t)}{dt} &= [(\$_{11} - \$_{21})(x - x^2) + (\$_{12} - \$_{22})(1 - 2x + x^2)] x(t) \\ &=: g(x) \end{aligned}$$

Im linken Panel dieser Vorlesung werden die Lösungen der Differentialgleichungen numerisch berechnet und analysiert. In Abhängigkeit der zugrundeliegenden Spielklasse des evolutionären Spiels sind unterschiedliche evolutionär stabile Strategien möglich (siehe [Evolutionär stabile Strategien](#)).

Folien der 4. Vorlesung

Vorlesungsaufzeichnung der 4. Vorlesung: [WS 2022/23](#) bzw. [WS 2021/22](#)

[View Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)

[Download Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)

[Download Python Programm: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele \(Version 1, Version 2\)](#)

[View Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)

[Download Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)

Physik der sozio-ökonomischen Systeme mit dem Computer

(Physics of Socio-Economic Systems with the Computer)

Vorlesung gehalten an der J.W.Goethe-Universität in Frankfurt am Main

(Sommersemester 2024)

von Dr.phil.nat. Dr.rer.pol. Matthias Hanauske

Frankfurt am Main 01.02.2024

Erster Vorlesungsteil:

Klassifizierung evolutionärer symmetrischer (2×2)-Spiele

Einführung

In diesem Unterkapitel werden die unterschiedlichen Spieltypen der gemischten Erweiterung eines simultanen (2 Spieler)-(2 Strategien) Spiels in strategischer Form mit symmetrischer Auszahlungsmatrix klassifiziert. Die Einteilung lehnt sich an das Buch von *Martin A. Nowak, Evolutionary Dynamics - Exploring the Equations of Life, 2006 (siehe Seite 49-51)* an. Eine alternative Klassifizierung findet man z.B. auch in *Matthias Hanauske, Evolutionäre Quanten-Spieltheorie im Kontext sozio-ökonomischer Systeme, 2011*. Ausgangspunkt ist die nebenstehende allgemeine symmetrische Auszahlungsmatrix eines (2 Personen)-(2 Strategien) Spiels (a, b, c und d sind reellwertige Zahlen). Da es sich um eine symmetrische Auszahlungsmatrix handelt, gilt: $\hat{\$}^B = (\hat{\$}^A)^T$.

Abhängig von den gewählten Parametern der Auszahlungsmatrix lassen sich symmetrische (2×2) Spiele in drei unterschiedliche Spielklassen gliedern: Dominante Spiele, Koordinationsspiele und Anti-Koordinationsspiele. Wir nehmen die folgende Form der Spielmatrix an:

$$\hat{\$}^A = \hat{\$} = \begin{pmatrix} \$_{11} & \$_{12} \\ \$_{21} & \$_{22} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (1)$$

Die Klasse der dominanten Spiele ($a > c$ und $b > d$ bzw. $a < c$ und $b < d$)

Numerisches Lösen der Differenzialgleichung

Näheres zum numerischen Lösen von Differenzialgleichungen für Studierende der Physik (Sommersemester 2022), speziell in Vorlesungen über Anfangswertproblemen und in dem Jupyter Notebook [Differenzialgleichungen](#) (Bemerkung: In den vergangenen Semestern benutzten wir noch die `odeint` Solver basiert.)

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
from scipy.integrate import solve_ivp
```

Definition der Funktion $g(x)$ und der Differenzialgleichung des evolutionären Spiels

```
def g(x,a,b,c,d):
    g = ((a-c)*(x-x*x) + (b-d)*(1-2*x+x*x))*x
    return g
```

```
def DGL(t, x):
    dxdt = g(x,a,b,c,d)
    return dxdt
```

Beispiel eines dominanten Spiels ($a=10 > c=7$)

Wir betrachten die zeitliche Entwicklung eines dominanten Spiels. $x(t)$ hängt neben der Funktion $g(x)$ von dem Anfangswert $x_0 := x(t=0)$ ab. Wenn wir z.B. von: 20% spielt Strategie 1 und 80% spielt Strategie 2) so kann man das Anfangswertproblem im Bereich $t \in [0, 5]$ unter Verwendung von `1000` Zeitschritten lösen:

```
a,b,c,d = 10,6,7,5
t_val = np.linspace(0, 5, 1000)
x0 = 0.2
Loes = solve_ivp(DGL, [0, 5], [x0], t_eval=t_val)
```

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib
4 from scipy.integrate import solve_ivp
5
6 # Definition der Funktion g
7 def g(x,a,b,c,d):
8     g = ((a-c)*(x-x*x) + (b-d)*(1-2*x+x*x))*x
9     return g
10
11 # Definition der DGL
12 def DGL(t, x):
13     dxdt = g(x,a,b,c,d)
14     return dxdt
15
16 # Groessenfestlegung der Labels usw. im Bild
17 params = {
18     'text.usetex'      : True,
19     'axes.titlesize'  : 22,
20     'axes.labelsize'  : 20,
21     'xtick.labelsize' : 20,
22     'ytick.labelsize' : 20
23 }
24 matplotlib.rcParams.update(params)
25
26 #Festlegung der Auszahlungsmatrix des symmetrischen (2x2)-Spiels
27 # Dominant Game
28 #a = -7
29 #b = -1
30 #c = -9
31 #d = -3
32 #tend=4
33 # Koordinantionsspiel
34 a = 2
35 b = 4
36 c = 0
37 d = 5
38 tend = 6
39 # Anti-Koordinantionsspiel
40 #a = -10
41 #b = 2
42 #c = 0
43 #d = 1
44 #tend = 3
45
46 #Weitere Festlegungen

```

Weiterführende Links

Folien der 4.Vorlesung

Vorlesungsaufzeichnung der 4.Vorlesung: [WS 2022/23](#) bzw. [WS 2021/22](#)

[View Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)

[Download Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)

[Download Python Programm: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele \(Version 1 , Version 2\)](#)

[View Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)

[Download Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)

[Download Python Programm: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele \(Version 1 , Version 2\)](#)

Auf der Internetseite der Vorlesung

Python Programm Version 2 (evol1a.py)

Zum Lösen des evolutionären Spiels mit Python werden wir in dieser Version die Funktion '[solve_ivp\(...\)](#)' verwenden, die sich im Untermodul '[scipy.integrate](#)' befindet, welche Funktionen zum Lösen von gewöhnlichen Differenzialgleichungen bereitstellt.

Lösen des evolutionären Spiels mit Python (Version 2)

```
46 #Weitere Festlegungen
47 numpoints = 500
48 t_val = np.linspace(0,tend,numpoints)
49 curvethick = 1.5
50
51 # Definition der Kurvenfarbe in Abhaengigkeit von der Spielklasse
52 # Dominant Games
53 ▼ if a > c and b > d:
54     cmap = plt.cm.Greys
55     texttitle = r'$\rm Dominantes\, Spiel $'
56 ▼ if a < c and b < d:
57     cmap = plt.cm.Greys
58     texttitle = r'$\rm Dominantes\, Spiel $'
59 # Koordinationsspiele
60 ▼ if a > c and b < d:
61     cmap = plt.cm.Blues
62     texttitle = r'$\rm Koordinationsspiel $'
63 # Anti-Koordinationsspiele
64 ▼ if a < c and b > d:
65     cmap = plt.cm.Oranges
66     texttitle = r'$\rm Anti-Koordinationsspiel $'
67
68 # Mehrere Anfangswerte der Population
69 num_x0 = 30
70 x0 = np.linspace(0,1,num_x0)
71 line_colors = cmap(np.linspace(0,1,num_x0+10))
72
73 # Loesung der DGL und plotten des Bildes
74 Loes = solve_ivp(DGL, [0, tend], x0, t_eval=t_val)
75 ▼ for j in range(num_x0):
76     plt.plot(Loes.t,Loes.y[j],c=line_colors[j+5], linewidth=curvethick, linestyle='-')
77
78 # Plotten der Spielmatrix in das Bild
79 textstr1 = r'$\hat{\bf {cal \}} = \left( \begin{array}{c}{c} a & b \\ c & d \end{array} \right) \backslash a='+str(a)+' , \backslash b='+str(b)+' $'
80 textstr2 = r'$\backslash c='+str(c)+' , \backslash d='+str(d)+' $'
81 props = dict(boxstyle='round', facecolor='white', alpha=0.92)
82 plt.text(tend-tend/50.0, 0.98, textstr1+textstr2, fontsize=16, verticalalignment='top', horizontalalignment='right', bbox=props)
83
84 # Achsenbeschriftungen usw.
85 plt.ylim(0,1)
86 plt.yticks([0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],["$0$", "", "$0.2$", "", "$0.4$", "", "$0.6$", "", "$0.8$", "", "$1$"])
87 plt.ylabel(r"$\rm x(t)$")
88 plt.xlabel(r"$\rm t$")
89 plt.title(texttitle)
90
91 #Speichern der Bilder als (.png , benoetigt dvi.png unter Linux)- und .pdf-Datei
92 saveFig = "./evol.png"
93 plt.savefig(saveFig, dpi=100, bbox_inches="tight", pad_inches=0.05, format="png")
94 saveFig = "./evol.pdf"
95 plt.savefig(saveFig, bbox_inches="tight", pad_inches=0.05, format="pdf")
96 plt.show()
97
```

$g_B(x, y)$ bestimmen (siehe [Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)).

Weiterführende Links

[Folien der 4. Vorlesung](#)

Vorlesungsaufzeichnung der 4. Vorlesung: [WS 2022/23](#) bzw. [WS 2021/22](#)

[View Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)

[Download Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)

[Download Python Programm: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele \(Version 1, Version 2\)](#)

[View Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)

[Download Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)

[Download Python Programm: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele \(Version 1, Version 2\)](#)

-
-
-
-
-
-
-
-
-

Auf der Internetseite der Vorlesung

Python Programm
Version 1 (evol1.py)

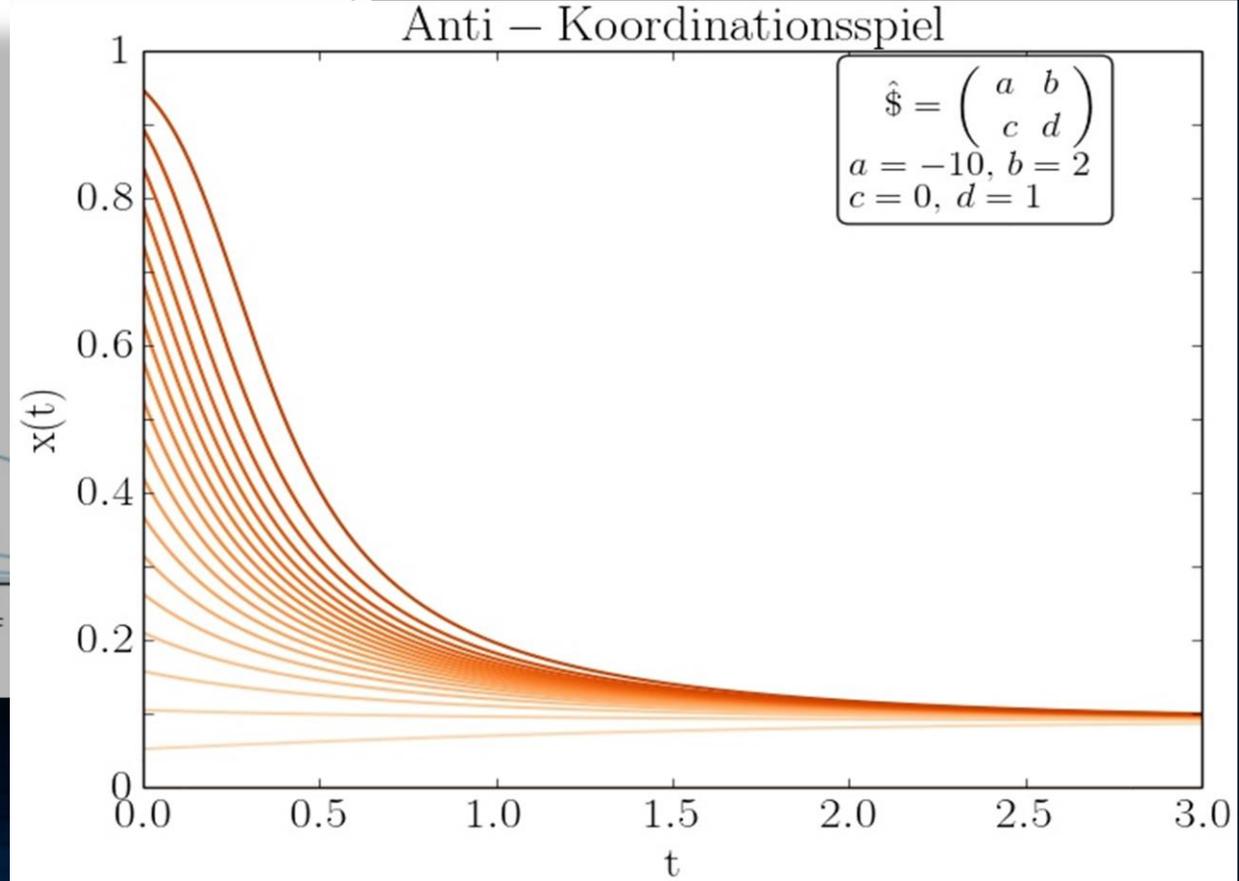
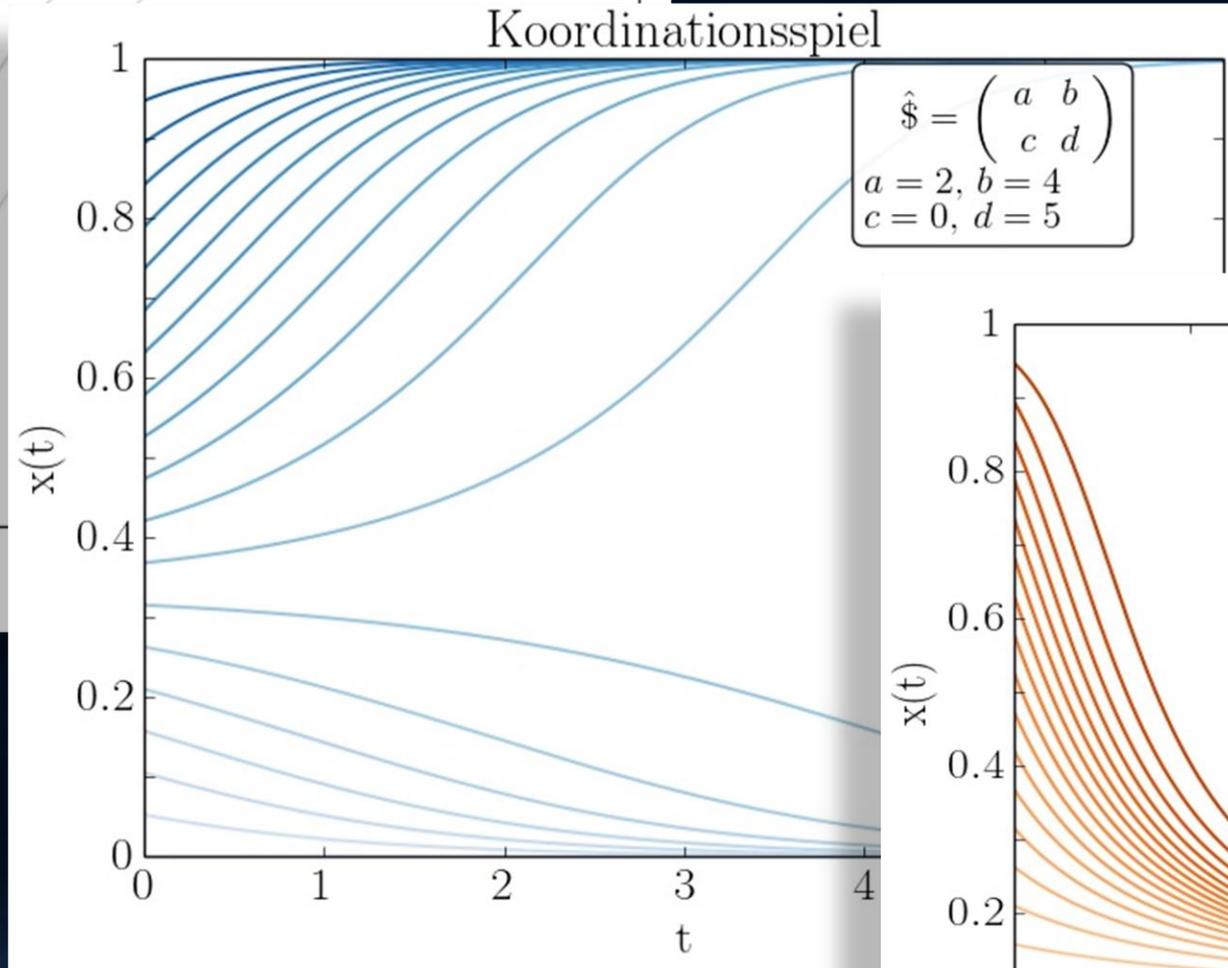
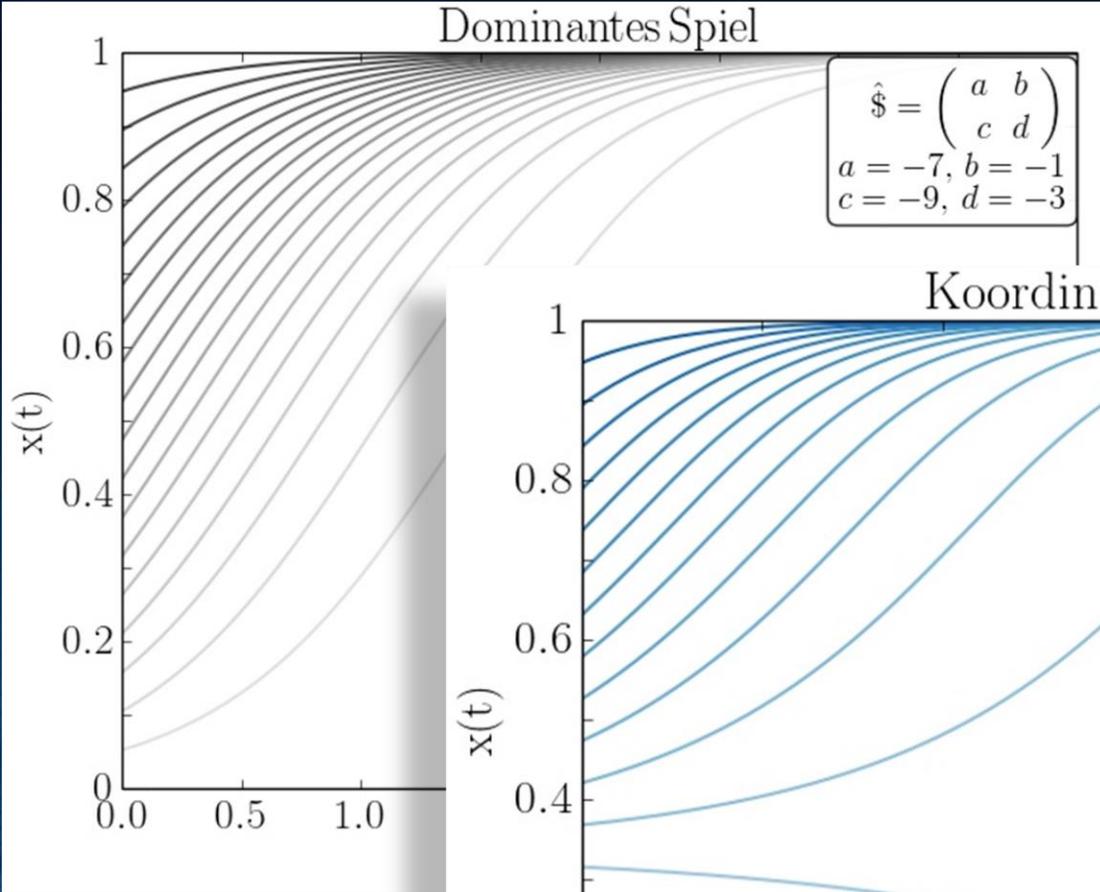
Lösen des evolutionären Spiels mit Python,
hier mit dem einfachen Euler-Verfahren

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib
4
5 # Definition der Funktion g
6 def g(x,a,b,c,d):
7     g = ((a-c)*(x-x*x) + (b-d)*(1-2*x+x*x))*x
8     return g
9
10 # Loesen der DGL
11 def solve_dgl(numpoints,tend,x0,a,b,c,d):
12     sol = np.empty([numpoints,2])
13     t = np.linspace(0,tend,numpoints)
14     dt = t[1]-t[0]
15     sol[0][0] = t[0]
16     sol[0][1] = x0
17     for i in range(1,numpoints):
18         sol[i][0] = t[i]
19         dx = g(sol[i-1,1],a,b,c,d)*dt
20         sol[i][1] = sol[i-1,1] + dx
21     return sol
22
23 # Groessenfestlegung der Labels usw. im Bild
24 params = {
25     'text.usetex' : True,
26     'axes.titlesize' : 22,
27     'axes.labelsize' : 20,
28     'xtick.labelsize' : 20,
29     'ytick.labelsize' : 20
30 }
31 matplotlib.rcParams.update(params)
32
33 #Festlegung der Auszahlungsmatrix des symmetrischen (2x2)-Spiels
34 # Dominant Game
35 #a = -7
36 #b = -1
37 #c = -9
38 #d = -3
39 #tend=4
40 # Koordinantionsspiel
41 a = 2
42 b = 4
43 c = 0
44 d = 5
45 tend = 6
46 # Anti-Koordinantionsspiel
47 #a = -10
48 #b = 2
49 #c = 0
50 #d = 1
51 #tend = 3
```

Lösen des evolutionären Spiels mit Python (Version 1)

```
53 #Weitere Festlegungen
54 numpoints = 500
55 curvethick = 1.5
56
57 # Definition der Kurvenfarbe in Abhaengigkeit von der Spielklasse
58 # Dominant Games
59 ▼ if a > c and b > d:
60     cmap = plt.cm.Greys
61     texttitle = r'$\rm Dominantes, Spiel $'
62 ▼ if a < c and b < d:
63     cmap = plt.cm.Greys
64     texttitle = r'$\rm Dominantes, Spiel $'
65 # Koordinationsspiele
66 ▼ if a > c and b < d:
67     cmap = plt.cm.Blues
68     texttitle = r'$\rm Koordinationsspiel $'
69 # Anti-Koordinationsspiele
70 ▼ if a < c and b > d:
71     cmap = plt.cm.Oranges
72     texttitle = r'$\rm Anti-Koordinationsspiel $'
73
74 # Mehrere Anfangswerte der Population
75 num_x0 = 30
76 x0 = np.linspace(0,1,num_x0)
77 line_colors = cmap(np.linspace(0,1,num_x0+10))
78
79 # Loesung der DGL und plotten des Bildes
80 ▼ for j in range(num_x0):
81     Loes = solve_dgl(numpoints,tend,x0[j],a,b,c,d)
82     plt.plot(Loes[:,0],Loes[:,1],c=line_colors[j+5], linewidth=curvethick, linestyle='-')
83
84 # Plotten der Spielmatrix in das Bild
85 textstr1 = r'$\hat{\bf {\cal \}} = \left( \begin{array}{cc} a & b \\ c & d \end{array} \right) \backslash a='+str(a)+'\, \backslash b='+str(b)+'$'
86 textstr2 = r'$\backslash c='+str(c)+'\, \backslash d='+str(d)+'$'
87 props = dict(boxstyle='round', facecolor='white', alpha=0.92)
88 plt.text(tend-tend/50.0, 0.98, textstr1+textstr2, fontsize=16, verticalalignment='top', horizontalalignment='right', bbox=props)
89
90 # Achsenbeschriftungen usw.
91 plt.ylim(0,1)
92 plt.yticks([0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],["$0$", "", "$0.2$", "", "$0.4$", "", "$0.6$", "", "$0.8$", "", "$1$"])
93 plt.ylabel(r"$\rm x(t)$")
94 plt.xlabel(r"$\rm t$")
95 plt.title(texttitle)
96
97
98 #Speichern der Bilder als (.png , benoetigt dvipng unter Linux)- und .pdf-Datei
99 saveFig = "./evol.png"
100 plt.savefig(saveFig, dpi=100, bbox_inches="tight", pad_inches=0.05, format="png")
101 saveFig = "./evol.pdf"
102 plt.savefig(saveFig, bbox_inches="tight", pad_inches=0.05, format="pdf")
103 plt.show()
```

Lösen des evolutionären Spiels mit Python Versionen 1 oder 2



E-Learning und interaktive Übungsaufgaben

Zusätzlich zu den Informationen aus dieser Internetseite finden Sie in diesem Unterpunkt diverse interaktive Übungsaufgaben zu den folgenden Themen:

Aufgabe 1

Reine Nash-Gleichgewichte in einem simultanen (2x2)-Spiel in strategischer Form mit symmetrischer Auszahlungsmatrix

Aufgabe 2

Gemischtes Nash-Gleichgewicht in einem simultanen (2x2)-Spiel in strategischer Form mit symmetrischer Auszahlungsmatrix

Aufgabe 3

Das gemischtes Nash-Gleichgewicht im Hirschjagt-Spiel

Aufgabe 4

Spielklassen von simultanen (2x2)-Spielen in strategischer Form mit symmetrischer Auszahlungsmatrix

Aufgabe 5

Zeitliche Entwicklung des Populationsvektors im evolutionären Spiel

Aufgabe 6

Evolutionär stabile Strategien

Aufgabe 7

Zeitliche Entwicklung des Populationsvektors im evolutionären Bi-Matrix Spiel

Aufgabe 8

Gemischtes Nash-Gleichgewicht in einem simultanen (2x2)-Spiel in strategischer Form mit unsymmetrischer Auszahlungsmatrix

Aufgabe 9

Gemischtes Nash-Gleichgewicht und zeitliche Entwicklung des Populationsvektors in Zentrumsspielen

Aufgabe 10

Zeitliche Entwicklung des Populationsvektors im evolutionären (2x3)-Spiel

Aufgabe 11

Gemischtes Nash-Gleichgewicht im evolutionären (2x3)-Spiel

Aufgabe 12

Mittlere Distanz zwischen zwei Knoten in einem zufälligen Netzwerk

Aufgabe 4

A/B	s_1	s_2
s_1	(107 , 107)	(10 , 177)
s_2	(177 , 10)	(125 , 125)

Betrachten Sie ein simultanes (2 Spieler)-(2 Strategien) Spiel in strategischer Form mit symmetrischer Auszahlungsmatrix. Die Menge der Spieler sei $\mathcal{I} = \{A, B\}$, die Menge der reinen Strategien sei $\mathcal{S}^A = \mathcal{S}^B = \{s_1, s_2\}$ und die Präferenzordnungen der Spieler sei durch die neben stehende Auszahlungstabelle quantifiziert.

Welcher Spielklasse gehört das Spiel an? Markieren Sie die zutreffende Aussage in der unteren Auswahl an

- Dominantes Spiel
- Koordinationsspiel
- Anti-Koordinationsspiel

und vergleichen Sie indem Sie den folgenden *Button* drücken.

Lösung anzeigen

Das Spiel gehört zur Klasse der

Aufgabe 5

Lösung

Der Wert des Populationsvektors zur Zeit $t=1.06$ ist $x(t=1.06) =$

[Weiter zur nächsten Aufgabe ...](#)

Betrachten Sie die zeitliche Entwicklung eines evolutionären (2x2)-Spiels mit symmetrischer Auszahlungsmatrix. Der Populationsvektors $\vec{x}(t)$ (hier $x(t)$, da nur zwei Strategien) wird durch folgende Differentialgleichung bestimmt

$$\frac{dx(t)}{dt} = [(\$_{11} - \$_{21})(x - x^2) + (\$_{12} - \$_{22})(1 - 2x + x^2)] x(t) := g(x)$$

$x(t)$, der Anteil der Spieler die zum Zeitpunkt t die Strategie s_1 spielen, hängt neben der Funktion $g(x)$ von dem Anfangswert $x(t=0)$ ab. Die Auszahlungswerte des Spiels seien $\$_{11} = 1$, $\$_{12} = 4$, $\$_{21} = 0$ und $\$_{22} = 2$ und der Anfangswert der Population zum Zeitpunkt $t=0$ sei $x(0) = 0.1$. Berechnen Sie den Wert des Populationsvektors zur Zeit $t=1.06$. Tragen Sie bitte Ihren Wert in das untere Eingabefeld ein

$x(t=1.06) =$

und vergleichen Sie indem Sie den folgenden *Button* drücken.

Lösung anzeigen

Aufgabe 6

Lösung

Die evolutionär stabile Strategie des Spiels lautet $x(t \rightarrow \infty) =$

[Weiter zur nächsten Aufgabe ...](#)

Betrachten Sie die zeitliche Entwicklung eines evolutionären (2x2)-Spiels mit symmetrischer Auszahlungsmatrix. Der Populationsvektors $\vec{x}(t)$ (hier $x(t)$, da nur zwei Strategien) wird durch folgende Differentialgleichung bestimmt

$$\frac{dx(t)}{dt} = [(\$_{11} - \$_{21})(x - x^2) + (\$_{12} - \$_{22})(1 - 2x + x^2)] x(t) := g(x)$$

$x(t)$, der Anteil der Spieler die zum Zeitpunkt t die Strategie s_1 spielen, hängt neben der Funktion $g(x)$ von dem Anfangswert $x(t=0)$ ab. Die Auszahlungswerte des Spiels seien $\$_{11} = 139$, $\$_{12} = 110$, $\$_{21} = 135$ und $\$_{22} = 145$ und der Anfangswert der Population zum Zeitpunkt $t=0$ sei $x(0) = 0.5$. Berechnen Sie die evolutionär stabile Strategie des Spiels; also gegen welchen Wert strebt der Populationsvektor für $t \rightarrow \infty$?

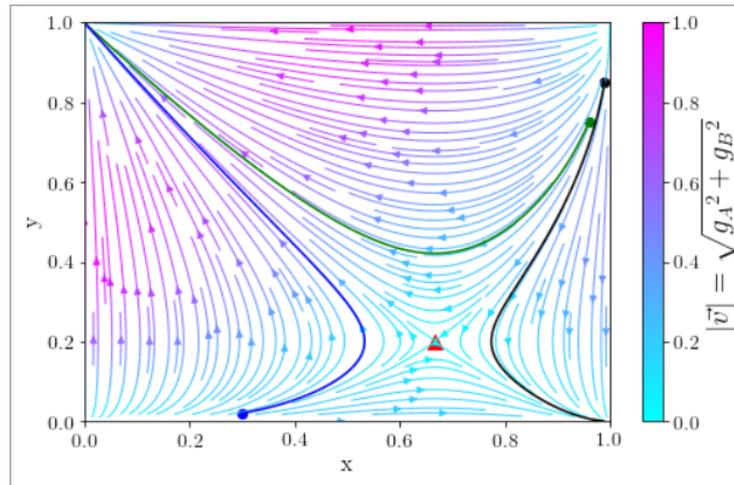
Tragen Sie bitte Ihren Wert in das untere Eingabefeld ein

$x(t \rightarrow \infty) =$

und vergleichen Sie indem Sie den folgenden *Button* drücken.

Lösung anzeigen

Evolutionäre unsymmetrische (2 × 2) Spiele (Bi-Matrix Spiele)



Die im oberen Bereich des rechten Panels dieser Vorlesung dargestellten Gleichungen der Reproduktionsdynamik lassen sich nach Spezifikation der Auszahlungsmatrizen $\hat{\A und $\hat{\B auf unterschiedlichste Populationsspiele anwenden. Generell lassen sich evolutionäre unsymmetrische (2 × 2) Spiele in die folgenden drei Spielklassen gliedern: Eckenspiele, Sattelpunktspiele und Zentrumsspiele. Ein Eckenspiel liegt vor, falls zumindest eine der Auszahlungsmatrizen der Spielergruppen eine dominante Struktur hat. Aufgrund der Dominanz der Strategie endet die zeitliche Entwicklung der Populationen, unabhängig von der Anfangsbedingungen, in einer Ecke des x-y Populationsvektor Diagramms - die ESS des Eckenspiels. Ein Sattelpunktspiel liegt vor, falls beide Spielergruppen gleichzeitig ein Koordinationsspiel oder Anti-Koordinationsspiel spielen. Bei einem Sattelpunktspiel das aus zwei Anti-Koordinationsspielen besteht, existieren zwei

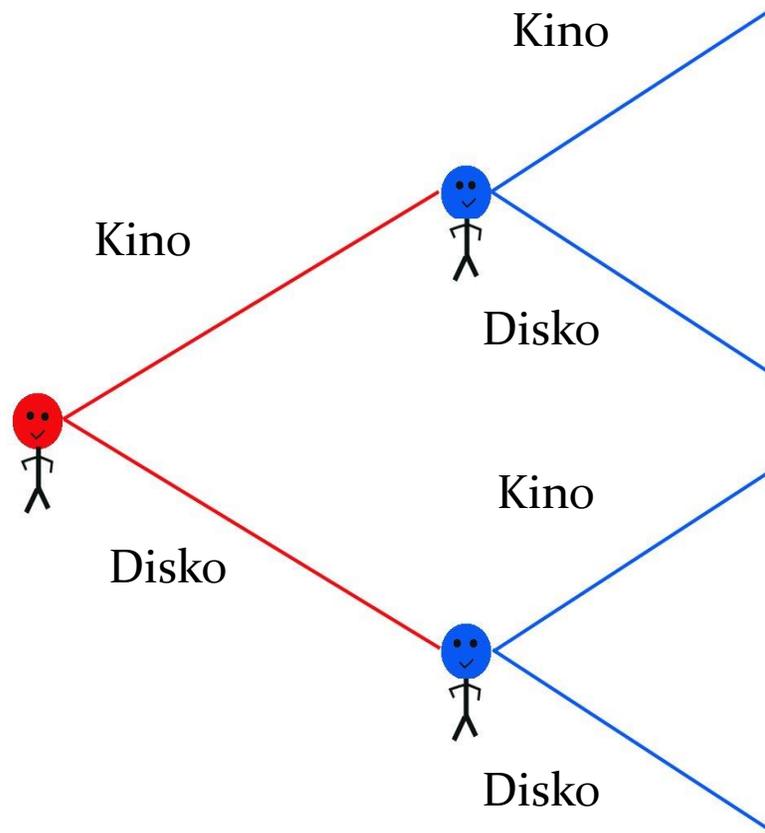
ESSs ((x=0,y=1) oder (x=1,y=0)) zu denen sich die Populationsgruppen im Laufe der Zeit entwickeln. Welche dieser evolutionär stabilen Strategien erreicht wird, hängt von der Anfangsstrategiewahl der Population ab. Die nebenstehende Abbildung zeigt die zeitliche Entwicklung von drei Anfangszuständen in einem solchen Eckenspiel. Auch bei sehr ähnlichen Werten der Anfangsstrategiewahl kann es geschehen, dass sich die Population im Laufe der Zeit zu unterschiedlichen Ecken entwickelt (siehe grüne und schwarze Trajektorien). Eine besondere Bedeutung hat der Sattelpunkt des Spiels (siehe rotes Dreieck). Die Position des Sattelpunktes im x-y Diagramm entspricht dem Wert des gemischten Nash-Gleichgewichtes bzw. lässt sich durch die Nullstellen der Funktionen $g_A(x, y)$ und $g_B(x, y)$ bestimmen. Bei einem Zentrumsspielen existiert keine ESS, da die Strategiewahl der Population sich im Laufe der Zeit ständig verändert und um ein Zentrum kreist. Die Position dieses Zentrums im x-y Diagramm entspricht dem Wert des gemischten Nash-Gleichgewichtes bzw. lässt sich durch die Nullstellen der Funktionen $g_A(x, y)$ und $g_B(x, y)$ bestimmen (siehe [Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)).

Weiterführende Links

- [Folien der 4. Vorlesung](#)
- Vorlesungsaufzeichnung der 4. Vorlesung: [WS 2022/23](#) bzw. [WS 2021/22](#)
- [View Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)
- [Download Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)
- Download Python Programm: Evolutionäre Spieltheorie symmetrischer (2x2)-Spiele ([Version 1](#) , [Version 2](#))
- [View Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)
- [Download Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)
- Download Python Programm: Evolutionäre Spieltheorie unsymmetrischer (2x2)-Spiele ([Version 1](#) , [Version 2](#))

Kampf der Geschlechter

	Kino	Disko
Kino	(1, 3)	(0, 0)
Disko	(0, 0)	(3, 1)



Alexander und Bettina haben bei ihrem letzten Treffen nicht genau ausgemacht, wann und wo sie sich am Samstagabend treffen wollen. Bettina geht sehr gerne in das kleine Kino am Stadtrand (Spätvorstellung, Beginn 23.30 Uhr), Alexander aber würde gerne in die Diskothek im Zentrum der Stadt gehen. Keiner von ihnen hat ein Telefon. Bettina denkt, wird er mir zuliebe ins Kino gehen? Alexander denkt ähnlich, wird sie mir zuliebe in die Diskothek gehen? Beiden liegt aber viel daran sich am Samstagabend zu treffen. Der erzielte Nutzen dieses Spiels kann zum Beispiel wie oben angegeben quantifiziert werden.

Kampf der Geschlechter

(Nash-Gleichgewichte in reinen Strategien)

- Es gibt keine dominante Strategie bei diesem Spiel.
- Es gibt zwei reine Nash-Gleichgewichte:

(Kino, Kino)

(Disko, Disko)

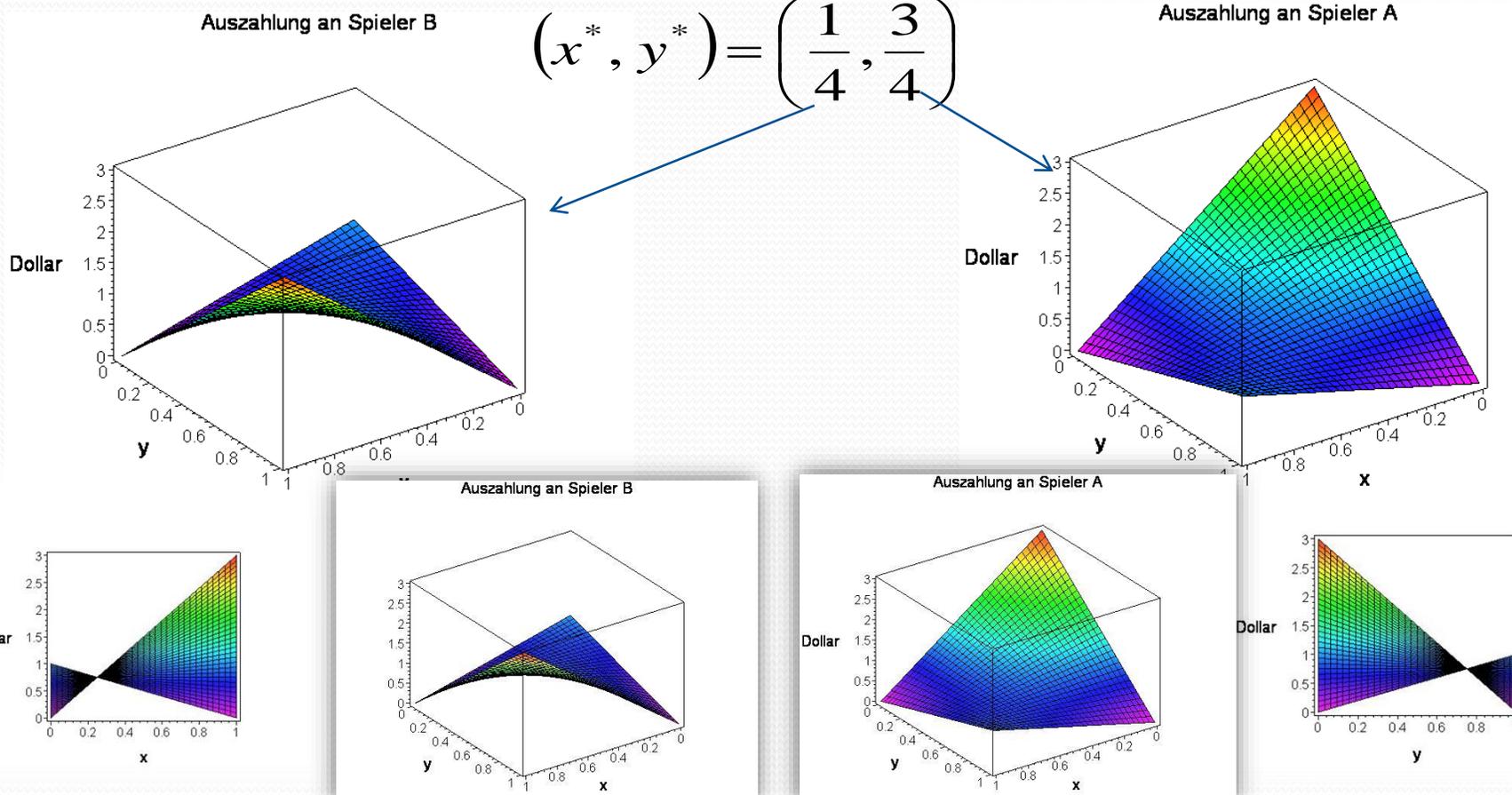
	Kino	Disko
Kino	(1, 3)	(0, 0)
Disko	(0, 0)	(3, 1)

Kampf der Geschlechter

(Grafische Veranschaulichung des gemischten Nash-Gleichgewichts)

- Das Nash-Gleichgewicht in gemischten Strategien befindet sich bei

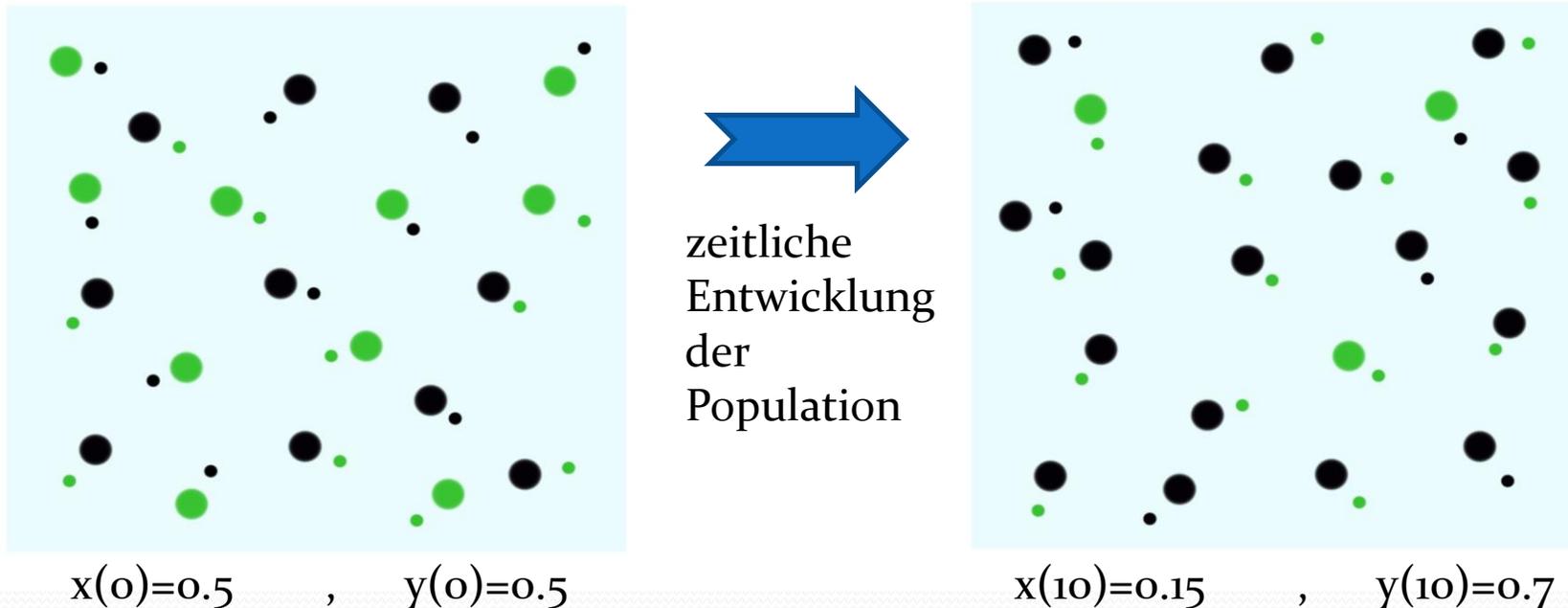
$$(x^*, y^*) = \left(\frac{1}{4}, \frac{3}{4} \right)$$



Evolutionäre Spieltheorie

Unsymmetrische (2x2)-Spiele (Bimatrixspiele) zweier Populationen

Bei unsymmetrischen (2x2)-Spiele besteht die zugrundeliegende Population aus zwei Gruppen (hier große und kleine Kreise). Aufgrund der unterschiedlichen Auszahlungsmatrizen können die Populationsgruppen sich in ihren Strategieentscheidungen (**grün**, schwarz) unterschiedlich entwickeln.



Mögliche Strategien: (**grün**, schwarz), Parameter t stellt die „Zeit“ dar.

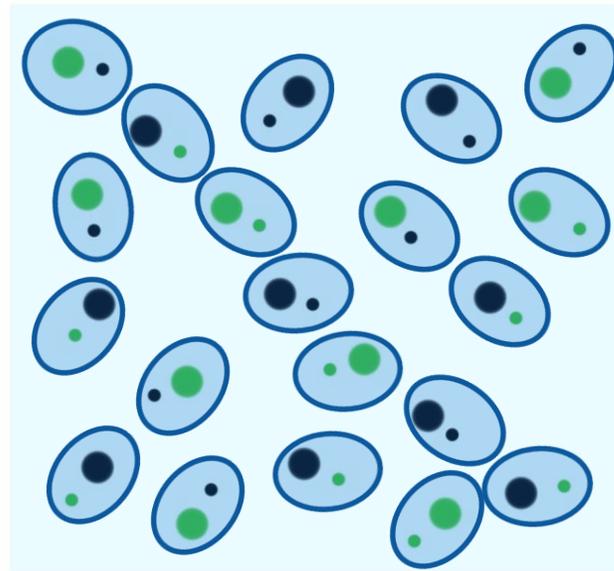
$x(t)$: Anteil der großen Spieler, die im Zeitpunkt t die Strategie „**grün**“ spielen.

$y(t)$: Anteil der kleinen Spieler, die im Zeitpunkt t die Strategie „**grün**“ spielen.

Evolutionäre Spieltheorie

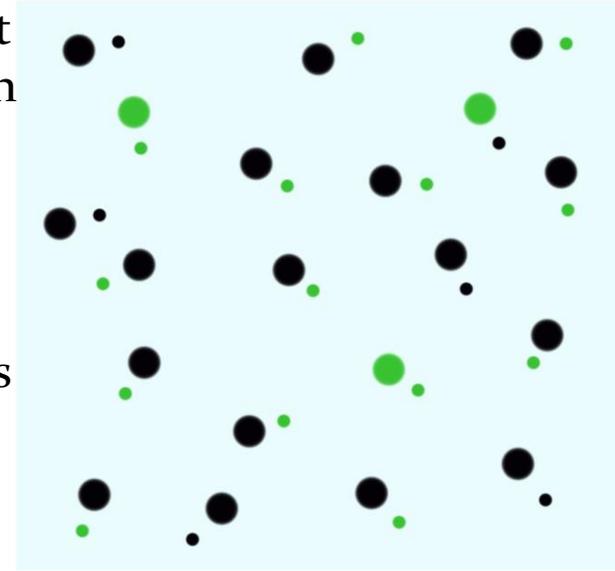
Unsymmetrische (2x2)-Spiele (Bimatrixspiele)

Die einzelnen Akteure innerhalb der betrachteten gesamten Population spielen ein andauernd sich wiederholendes Spiel miteinander, wobei sich jeweils zwei Spieler mit unterschiedlichen Gruppenzugehörigkeiten zufällig treffen, das Spiel spielen und danach zu dem nächsten Spielpartner der anderen Gruppe wechseln.



$$x(10)=0.5, \quad y(10)=0.5$$

Die Anfangspopulation von Spielern spielt zum Zeitpunkt $t=0$ das erste Mal das Spiel. Es bilden sich stets Zweier-Gruppen aus großen und kleinen Kreisen.



$$x(10)=0.15, \quad y(10)=0.7$$

Das evolutionäre Spiel schreitet voran und die **grüne** Strategie wird für die kleinen Spieler zunehmend attraktiver ($y(10)=0.7$), wohingegen sie für die großen Spieler zunehmend weniger attraktiv wird ($x(10)=0.15$).

Wir beschränken uns im folgenden auf den 2-Strategien Fall ($m_A = m_B = 2$), lassen jedoch weiter eine Unsymmetrie der Auszahlungsmatrix zu. Die beiden Komponenten der zweidimensionalen gruppenspezifischen Populationsvektoren lassen sich dann, aufgrund ihrer Normalisierungsbedingung, auf eine Komponente reduzieren ($x_2^A = 1 - x_1^A$ und $x_2^B = 1 - x_1^B$). Das zeitliche Verhalten der Komponenten der Populationsvektoren (Gruppe A: $x(t) := x_1^A(t)$ und Gruppe B: $y(t) := x_1^B(t)$) wird in der Reproduktionsdynamik mittels des folgenden Systems von Differentialgleichungen beschrieben (siehe z.B. [4], S:116 oder [3], S:69):

$$\frac{dx(t)}{dt} = [(\beta_{11}^A + \beta_{22}^A - \beta_{12}^A - \beta_{21}^A) y(t) + (\beta_{12}^A - \beta_{22}^A)] (x(t) - (x(t))^2) =: g_A(x, y) \quad (2)$$

$$\frac{dy(t)}{dt} = [(\beta_{11}^B + \beta_{22}^B - \beta_{12}^B - \beta_{21}^B) x(t) + (\beta_{12}^B - \beta_{22}^B)] (y(t) - (y(t))^2) =: g_B(x, y)$$

Beispiel 1:

Kampf der Geschlechter als evolutionäres Spiel

Das gekoppelte System von Differentialgleichung lautet:

$$\frac{dx(t)}{dt} = x(t) \cdot (4 \cdot y(t) - 4 \cdot x(t) \cdot y(t) + 3 \cdot x(t) - 3)$$
$$\frac{dy(t)}{dt} = y(t) \cdot (4 \cdot x(t) - 4 \cdot x(t) \cdot y(t) + y(t) - 1)$$

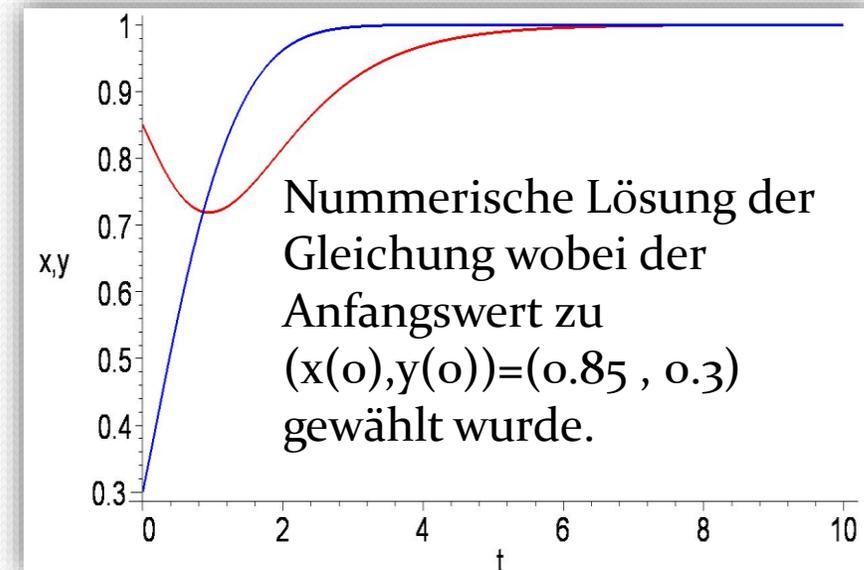
Die beiden Gruppen der Population sind Männer (A, $x(t)$) und Frauen (B, $y(t)$). Da nur zwei Strategien wählbar sind, lassen sich die jeweiligen Populationsanteile durch lediglich eine Größe ausdrücken ($x(t)$ und $y(t)$).

$$x(t) := x_1^A(t) \quad , \quad x_2^A(t) = 1 - x(t)$$

$$y(t) := x_1^B(t) \quad , \quad x_2^B(t) = 1 - y(t)$$

	Kino	Disko
Kino	(1, 3)	(0, 0)
Disko	(0, 0)	(3, 1)

Die Lösung der Gleichung erfolgt wiederum durch Integration bzw. kann mithilfe des Computers numerisch berechnet werden. Es muss in beiden Fällen ein fester Anfangswert $(x(0), y(0))$ gewählt werden.



Bi-Matrix Spiele

Gemischte Nash-Gleichgewichte

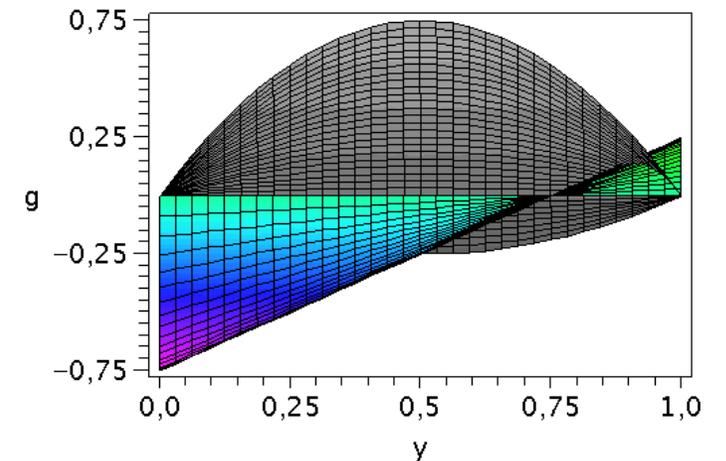
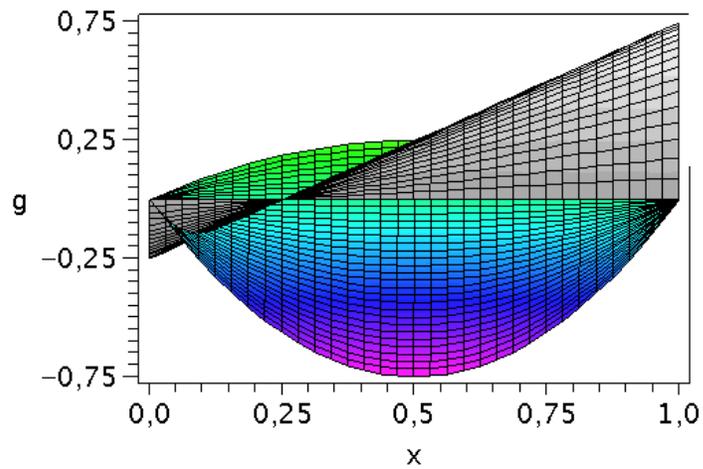
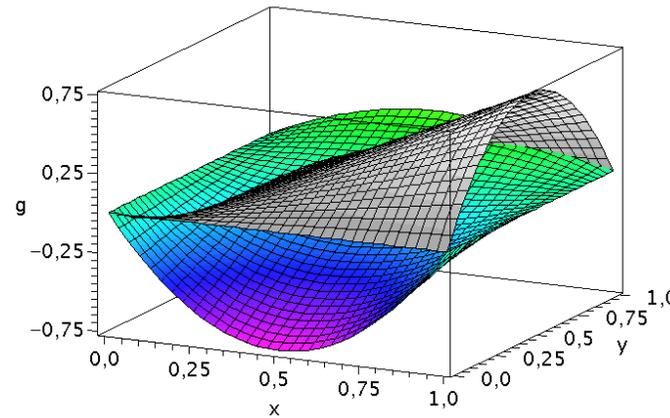
Nash-Gleichgewicht in gemischten Strategien:

$$\left. \frac{\partial \tilde{\$}^A(\tilde{s}^A, \tilde{s}^B)}{\partial \tilde{s}^A} \right|_{\tilde{s}^B = \tilde{s}^{B*}} = 0 \quad \forall \tilde{s}^A \in [0, 1] \quad , \quad \tilde{s}^{B*} \in]0, 1[$$

$$\left. \frac{\partial \tilde{\$}^B(\tilde{s}^A, \tilde{s}^B)}{\partial \tilde{s}^B} \right|_{\tilde{s}^A = \tilde{s}^{A*}} = 0 \quad \forall \tilde{s}^B \in [0, 1] \quad , \quad \tilde{s}^{A*} \in]0, 1[$$

Eigenschaften der Funktionen $g_A(x,y)$ und $g_B(x,y)$

Die das Bimatrix Spiel bestimmenden Funktionen $g_A(x,y)$ (farbige Fläche) und $g_B(x,y)$ (graue Fläche) sind in den unteren Abbildungen veranschaulicht. Beide Teilgruppenspiele gehören der Klasse der Koordinationsspiele an.



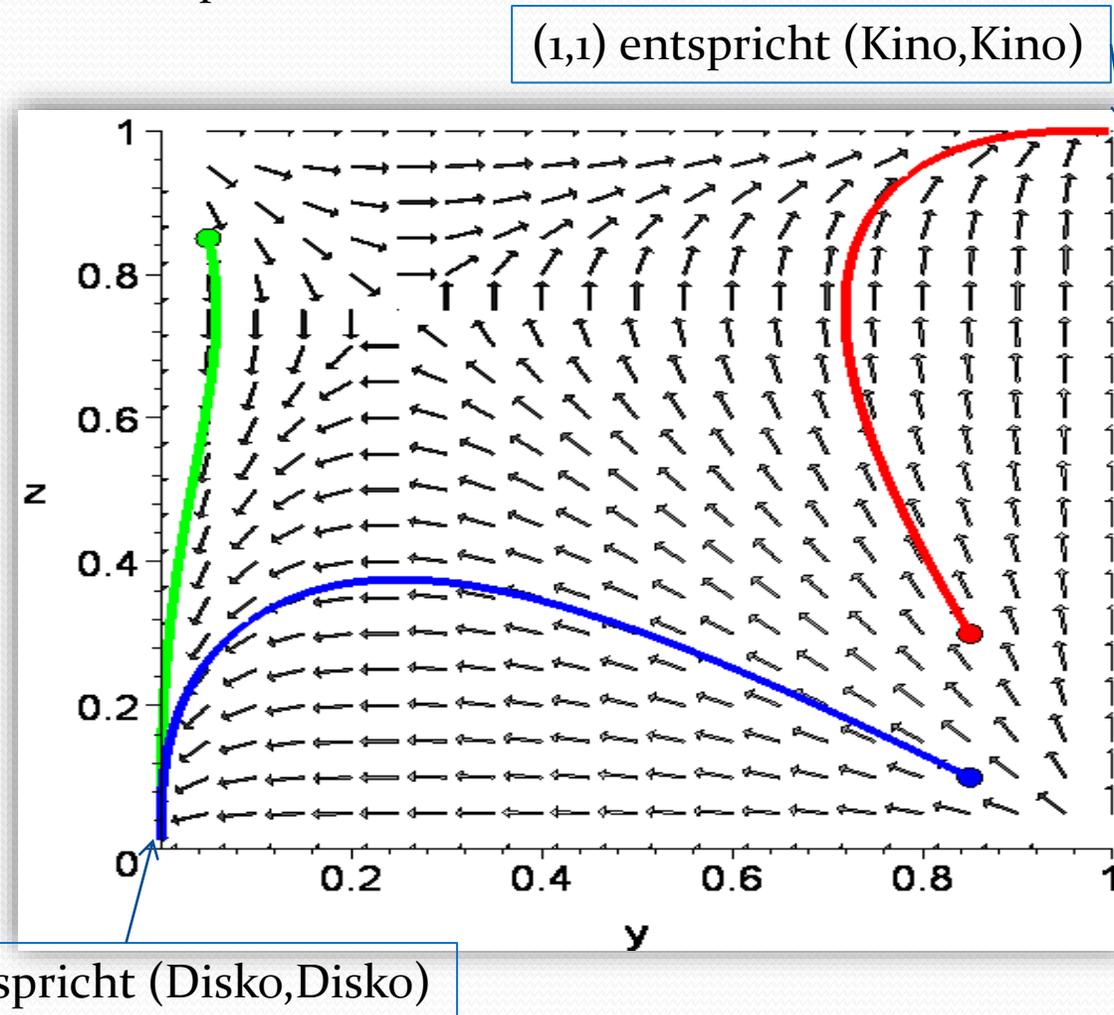
Beispiel 1:

Kampf der Geschlechter als evolutionäres Spiel

	Kino	Disko
Kino	(1, 3)	(0, 0)
Disko	(0, 0)	(3, 1)

Das „Kampf der Geschlechter“- Spiel gehört der Klasse der „Sattelpunktspiele“ (Saddle Class) an. Das Phasenportrait des Spiels besitzt das folgende Aussehen:

Die rechte Abbildung zeigt das zeitliche Verhalten von drei numerischen Lösungen mit unterschiedlichen Anfangsbedingungen. Die evolutionäre Erweiterung des Spiels besitzt zwei evolutionär stabile Strategien ((0,0) und (1,1)). Die blaue und grüne Populationsentwicklung enden bei (0,0) während die Anfangsbedingung der roten Population bei (1,1) endet.



Beispiel 2:

Klasse der Zentrumsspiele (Center Class)

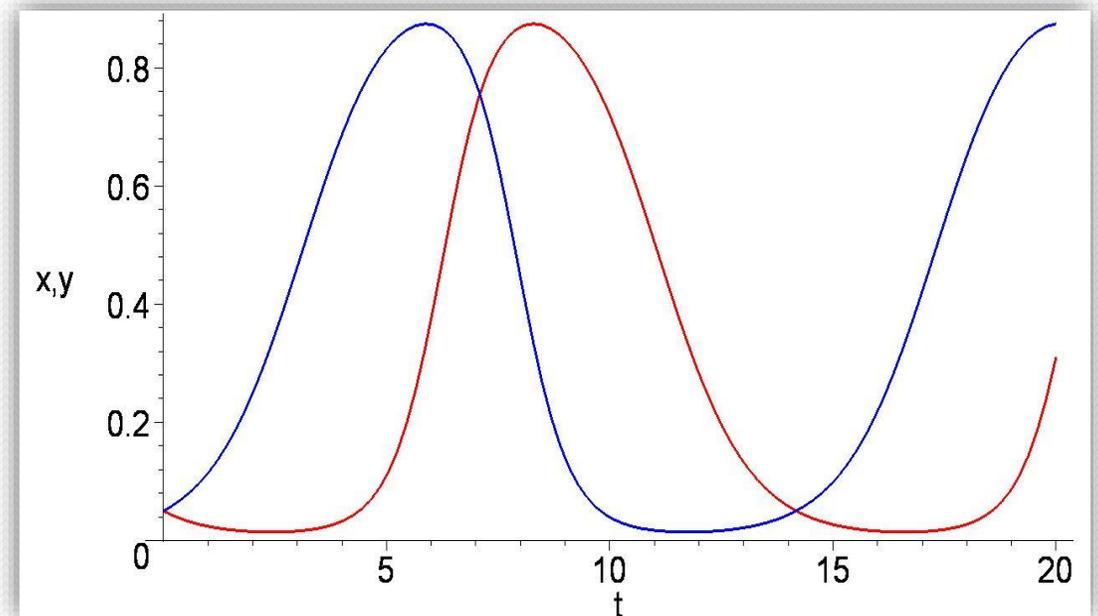
Das gekoppelte System von Differentialgleichung für dieses Beispiel lautet:

$$\frac{dx(t)}{dt} = x(t) \cdot (3 \cdot y(t) - 3 \cdot x(t) \cdot y(t) + x(t) - 1)$$
$$\frac{dy(t)}{dt} = y(t) \cdot (-3 \cdot x(t) + 3 \cdot x(t) \cdot y(t) - y(t) + 1)$$

Die rechte Abbildung zeigt eine numerische Lösung der obigen Gleichung, wobei der Anfangswert zu $(x(0), y(0)) = (0.05, 0.05)$ gewählt wurde. Im Gegensatz zu allen anderen möglichen Klassen von Bimatrixspielen, treten bei der Klasse der Zentrumsspiele periodische Verläufe der Populationsanteile $x(t)$ und $y(t)$ auf - die Populationsanteile enden nicht in einer evolutionär stabilen Strategie.

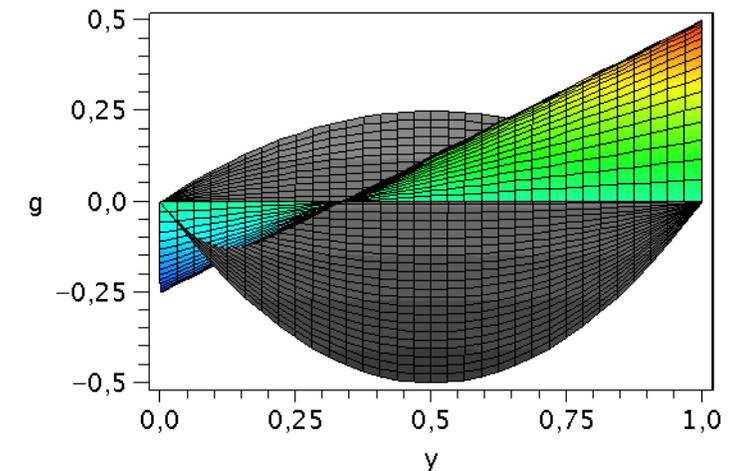
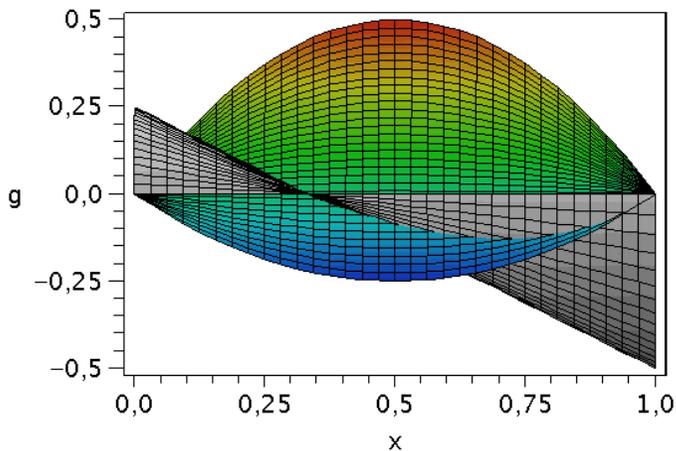
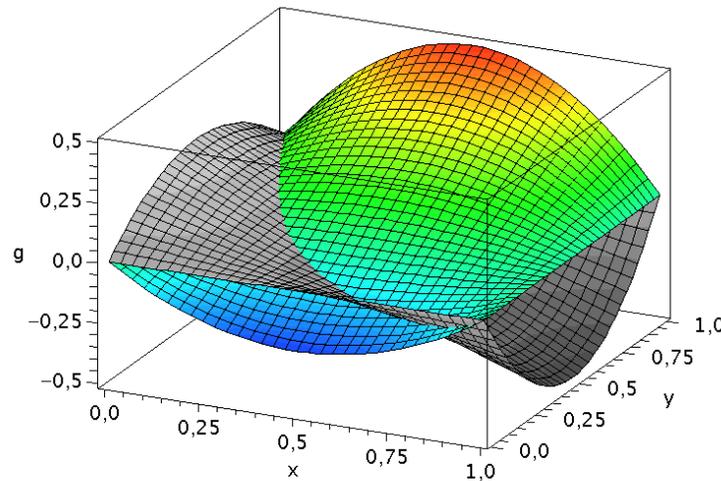
	Strat. 1	Strat. 2
Strat. 1	(2, -2)	(0, 0)
Strat. 2	(0, 0)	(1, -1)

Die Lösung der Gleichung erfolgt wiederum durch Integration bzw. kann mithilfe des Computers numerisch berechnet werden. Es wurde der Anfangswert zu $(x(0), y(0)) = (0.05, 0.05)$ gewählt.



Eigenschaften der Funktionen $g_A(x,y)$ und $g_B(x,y)$

Die das Bimatrix Spiel bestimmenden Funktionen $g_A(x,y)$ (farbige Fläche) und $g_B(x,y)$ (graue Fläche) sind in den unteren Abbildungen veranschaulicht. Das Spiel der Gruppe A gehört der Klasse der Koordinationsspiele an, das der Gruppe B der Klasse der Anti-Koordinationsspiele.



Beispiel 2:

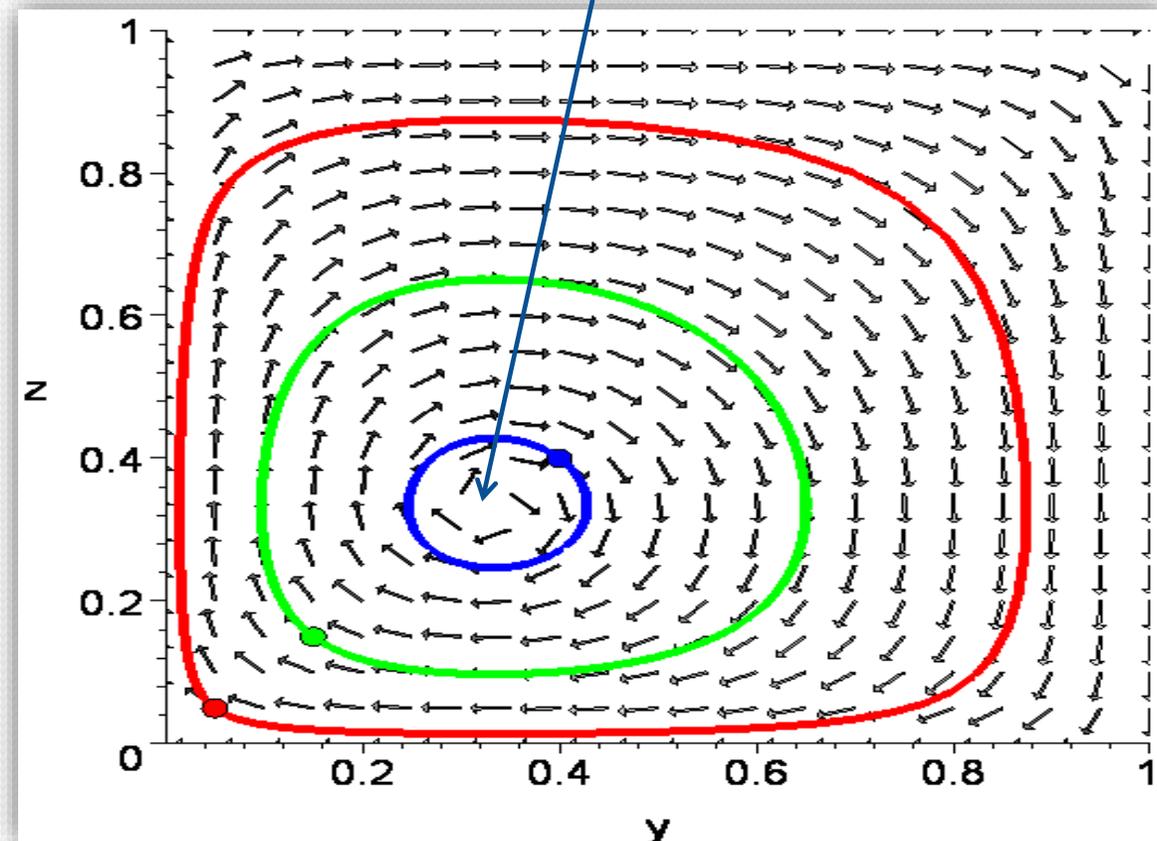
Klasse der Zentrumsspiele (Center Class)

	Strat. 1	Strat. 2
Strat. 1	(2, -2)	(0, 0)
Strat. 2	(0, 0)	(1, -1)

Das Phasenportrait des zweiten Beispiels besitzt das folgende Aussehen:

Die rechte Abbildung zeigt das zeitliche Verhalten von drei numerischen Lösungen mit unterschiedlichen Anfangsbedingungen. Dieses Bimatrixspiel besitzt keine evolutionär stabile Strategie, da die einzelnen Phasenraum-Trajektorien sich keinem Punkt annähern, sondern auf einer geschlossenen, zyklischen Bahn um das gemischte Nash-Gleichgewicht kreisen.

Zentrum: Gemischtes Nash-Gleichgewicht des Spiels



Beispiel 3:

Klasse der Eckenspiele (Corner Class)

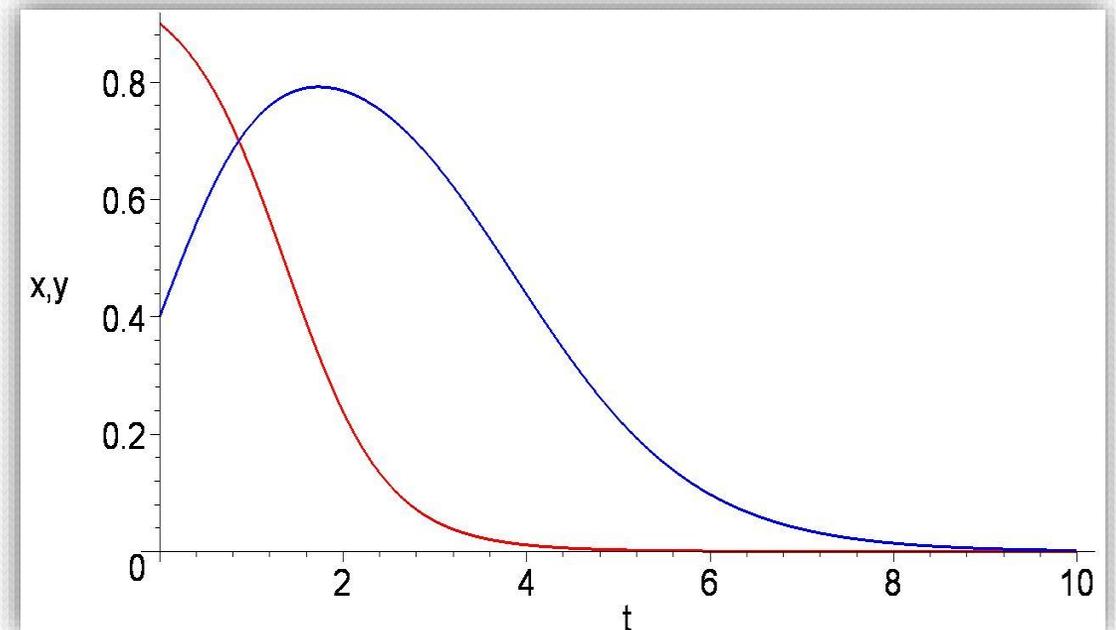
	Strat. 1	Strat. 2
Strat. 1	(-2, 2)	(0, 0)
Strat. 2	(0, 0)	(1, 1)

Das gekoppelte System von Differentialgleichung für dieses Beispiel lautet:

$$\frac{dx(t)}{dt} = x(t) \cdot (-y(t) + x(t) \cdot y(t) + x(t) - 1)$$
$$\frac{dy(t)}{dt} = y(t) \cdot (3 \cdot x(t) - 3 \cdot x(t) \cdot y(t) + y(t) - 1)$$

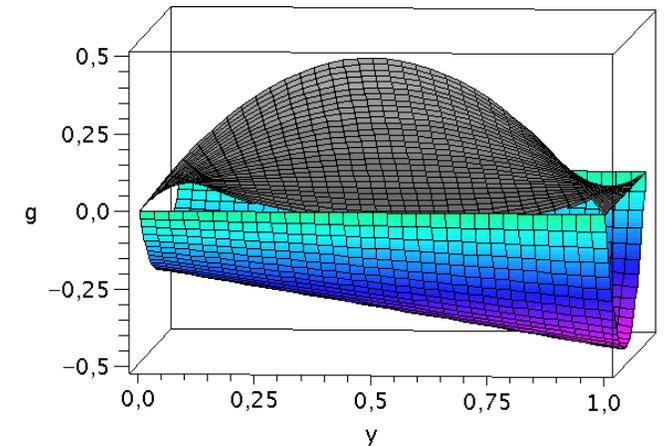
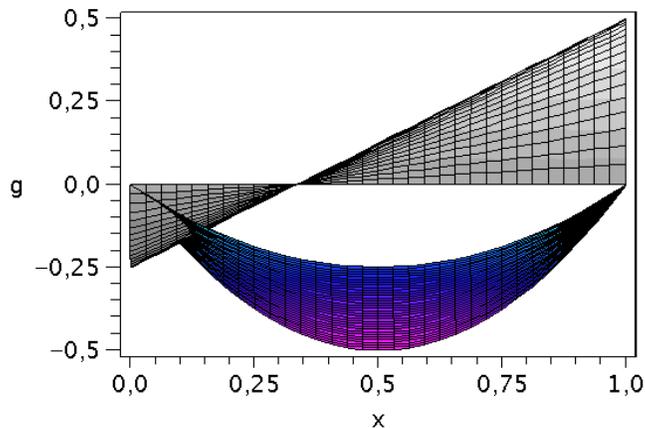
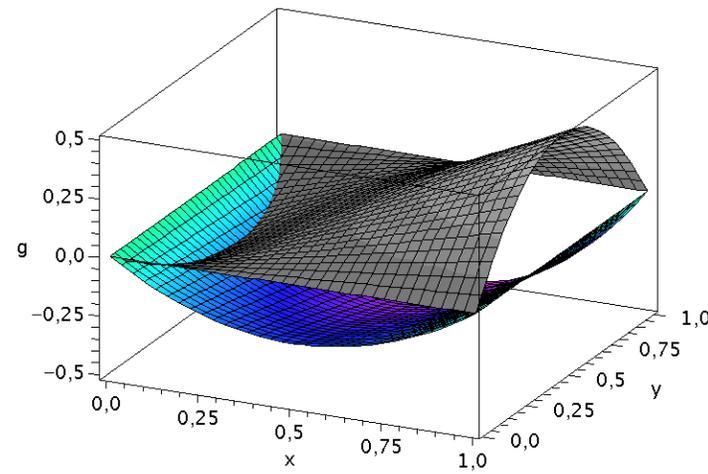
Die Lösung der Gleichung erfolgt wiederum durch Integration bzw. kann mithilfe des Computers numerisch berechnet werden. Es wurde der Anfangswert zu $(x(0), y(0)) = (0.9, 0.4)$ gewählt.

Die rechte Abbildung zeigt eine numerische Lösung der obigen Gleichung, wobei der Anfangswert zu $(x(0), y(0)) = (0.9, 0.4)$ gewählt wurde. Bei der Klasse der „Eckspiele“ gibt es eine evolutionär stabile Strategie, die unabhängig von der gewählten Anfangsbedingung stets von der Population angestrebt wird.



Eigenschaften der Funktionen $g_A(x,y)$ und $g_B(x,y)$

Die das Bimatrix Spiel bestimmenden Funktionen $g_A(x,y)$ (farbige Fläche) und $g_B(x,y)$ (graue Fläche) sind in den unteren Abbildungen veranschaulicht. Das Spiel der Gruppe A gehört der Klasse der dominanten Spiele an, das der Gruppe B der Klasse der Koordinationsspiele.



Beispiel 3:

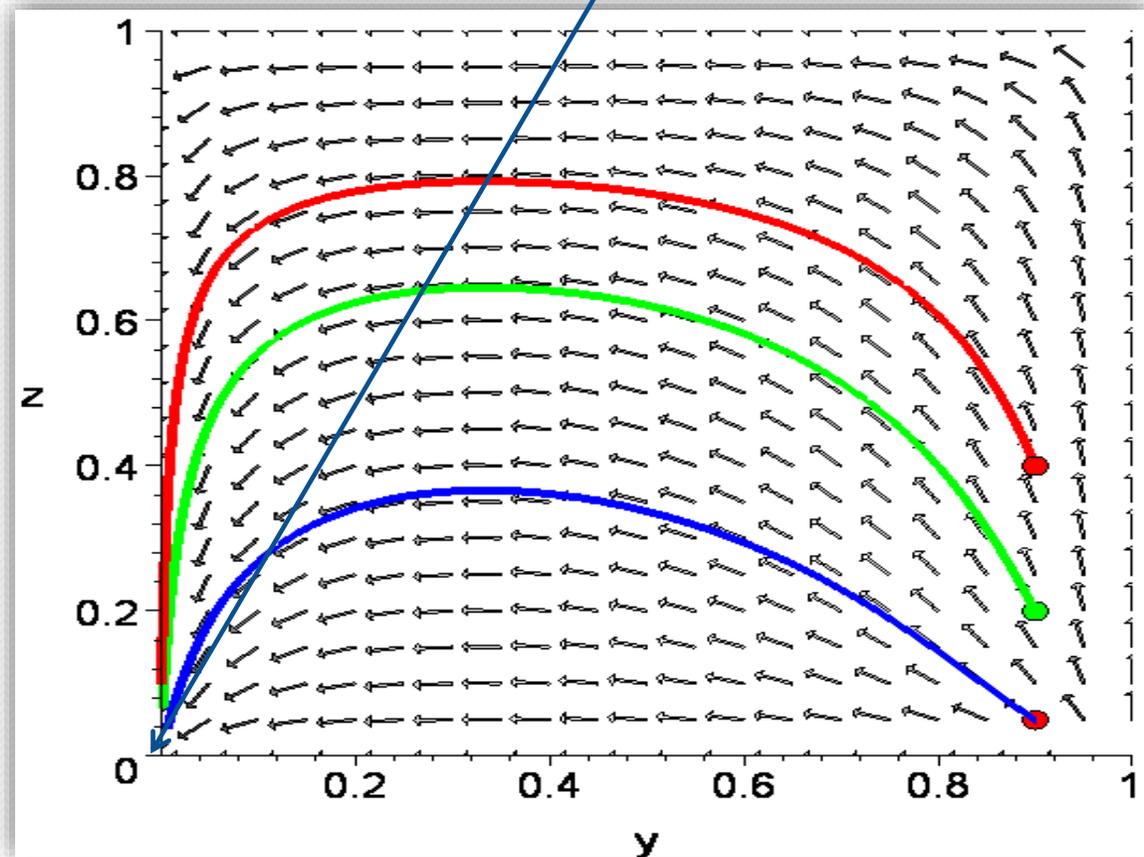
Klasse der Eckspiele (Corner Class)

Das Phasenportrait des dritten Beispiels besitzt das folgende Aussehen:

	Strat. 1	Strat. 2
Strat. 1	$(-2, 2)$	$(0, 0)$
Strat. 2	$(0, 0)$	$(1, 1)$

Die rechte Abbildung zeigt das zeitliche Verhalten von drei numerischen Lösungen mit unterschiedlichen Anfangsbedingungen. Dieses Bimatrixspiel besitzt eine evolutionär stabile Strategie, da es nur ein gemeinsames symmetrisches Nash-Gleichgewicht gibt $((x,y)=(0,0))$. Der rote Spieler besitzt sogar bei $(0,0)$ eine dominante Strategie.

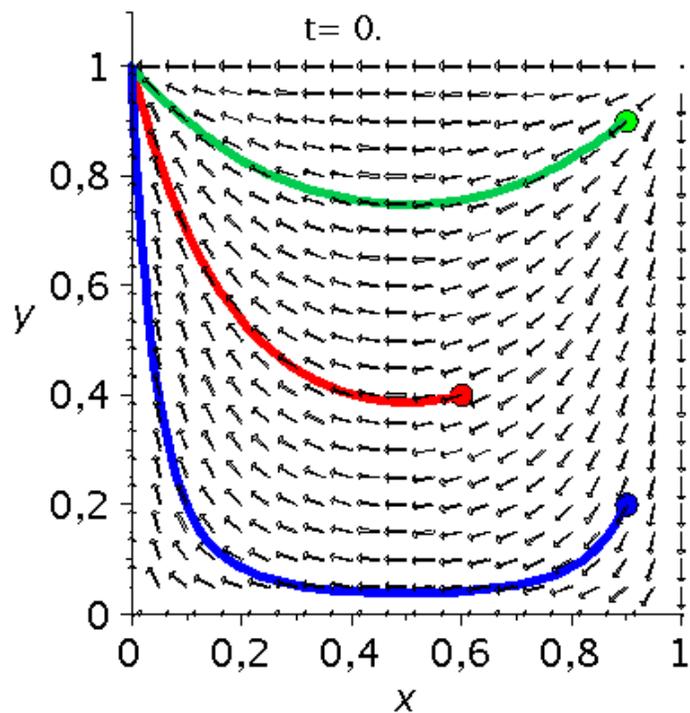
Einziges gemeinsames Nash-Gleichgewicht des Spiels



Klassifizierung von Bi-Matrix Spielen

Eckspiele

Die Spielklasse der Gruppe A oder der Gruppe B ist ein Dominantes Spiel



Sattelspiele

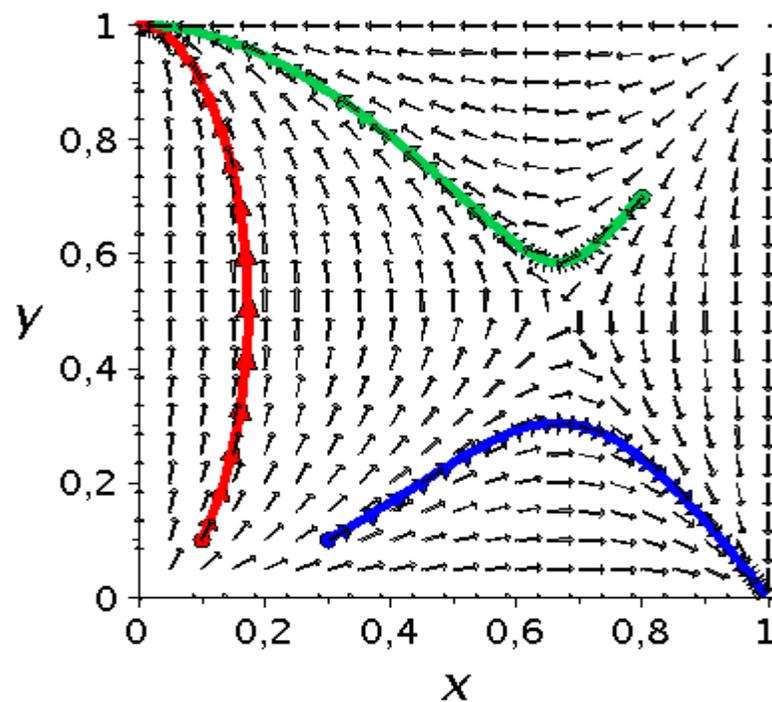
Spiel A: Koordinationsspiel

Spiel B: Koordinationsspiel

oder

Spiel A: Anti-Koordinationsspiel

Spiel B: Anti-Koordinationsspiel



Zentrumsspiele

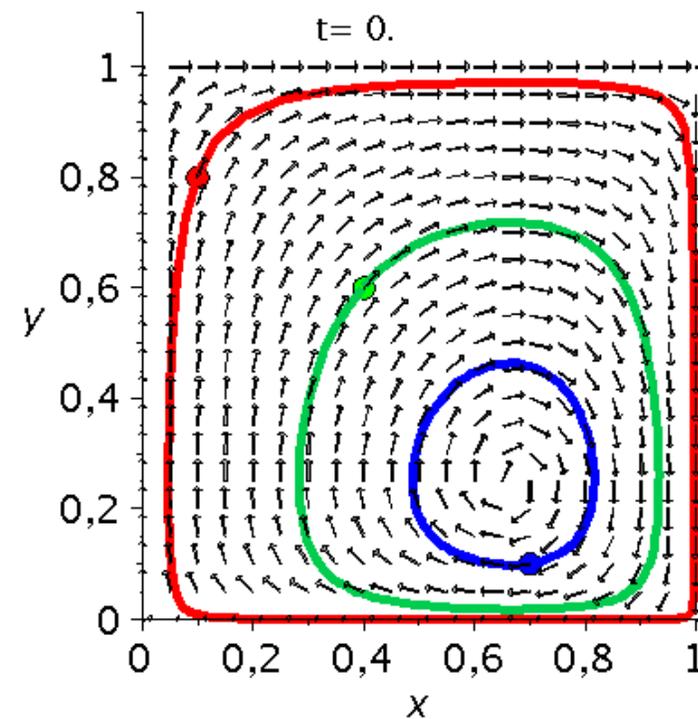
Spiel A: Koordinationsspiel

Spiel B: Anti-Koordinationsspiel

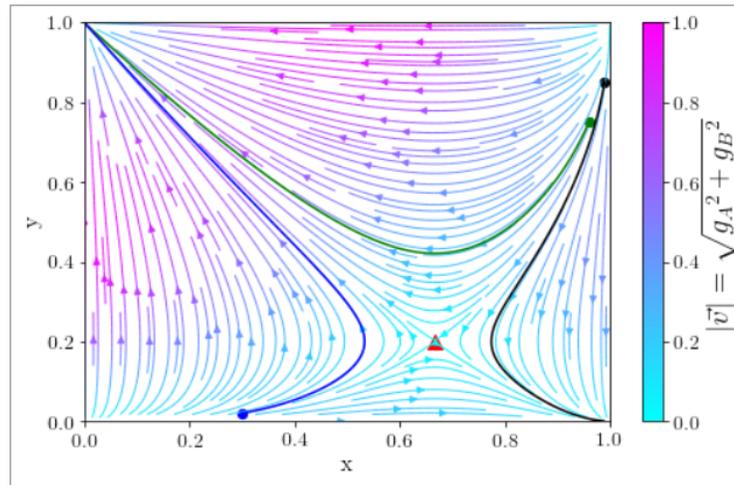
oder

Spiel A: Anti-Koordinationsspiel

Spiel B: Koordinationsspiel



Evolutionäre unsymmetrische (2 × 2) Spiele (Bi-Matrix Spiele)



Die im oberen Bereich des rechten Panels dieser Vorlesung dargestellten Gleichungen der Reproduktionsdynamik lassen sich nach Spezifikation der Auszahlungsmatrizen $\hat{\A und $\hat{\B auf unterschiedlichste Populationsspiele anwenden. Generell lassen sich evolutionäre unsymmetrische (2 × 2) Spiele in die folgenden drei Spielklassen gliedern: Eckenspiele, Sattelpunktspiele und Zentrumsspiele. Ein Eckenspiel liegt vor, falls zumindest eine der Auszahlungsmatrizen der Spielergruppen eine dominante Struktur hat. Aufgrund der Dominanz der Strategie endet die zeitliche Entwicklung der Populationen, unabhängig von der Anfangsbedingungen, in einer Ecke des x-y Populationsvektor Diagramms - die ESS des Eckenspiels. Ein Sattelpunktspiel liegt vor, falls beide Spielergruppen gleichzeitig ein Koordinationsspiel oder Anti-Koordinationsspiel spielen. Bei einem Sattelpunktspiel das aus zwei Anti-Koordinationsspielen besteht, existieren zwei

ESSs (($x=0, y=1$) oder ($x=1, y=0$)) zu denen sich die Populationsgruppen im Laufe der Zeit entwickeln. Welche dieser evolutionär stabilen Strategien erreicht wird, hängt von der Anfangsstrategiewahl der Population ab. Die nebenstehende Abbildung zeigt die zeitliche Entwicklung von drei Anfangszuständen in einem solchen Eckenspiel. Auch bei sehr ähnlichen Werten der Anfangsstrategiewahl kann es geschehen, dass sich die Population im Laufe der Zeit zu unterschiedlichen Ecken entwickelt (siehe grüne und schwarze Trajektorien). Eine besondere Bedeutung hat der Sattelpunkt des Spiels (siehe rotes Dreieck). Die Position des Sattelpunktes im x-y Diagramm entspricht dem Wert des gemischten Nash-Gleichgewichtes bzw. lässt sich durch die Nullstellen der Funktionen $g_A(x, y)$ und $g_B(x, y)$ bestimmen. Bei einem Zentrumsspielen existiert keine ESS, da die Strategiewahl der Population sich im Laufe der Zeit ständig verändert und um ein Zentrum kreist. Die Position dieses Zentrums im x-y Diagramm entspricht dem Wert des gemischten Nash-Gleichgewichtes bzw. lässt sich durch die Nullstellen der Funktionen $g_A(x, y)$ und $g_B(x, y)$ bestimmen (siehe [Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)).

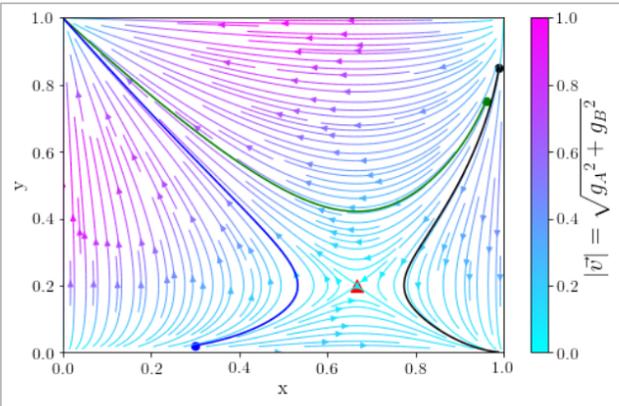
Weiterführende Links

- [Folien der 4. Vorlesung](#)
- Vorlesungsaufzeichnung der 4. Vorlesung: [WS 2022/23](#) bzw. [WS 2021/22](#)
- [View Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)
- [Download Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)
- Download Python Programm: Evolutionäre Spieltheorie symmetrischer (2x2)-Spiele ([Version 1](#) , [Version 2](#))
- [View Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)
- [Download Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)
- Download Python Programm: Evolutionäre Spieltheorie unsymmetrischer (2x2)-Spiele ([Version 1](#) , [Version 2](#))

Jupyter Notebook

Auf der Internetseite der Vorlesung

Evolutionäre unsymmetrische (2 × 2) Spiele (Bi-Matrix S



Die im oberen Bereich dargestellten Gleichung sich nach Spezifikation auf unterschiedlichste P lassen sich evolutionäre folgenden drei Spi Sattelpunktspiele und Z vor, falls zumindest i Spielergruppen eine de Dominanz der Strategie Populationen, unabhän; einer Ecke des x-y Popi des Eckenspiels. Ein S Spielergruppen gleiche Koordinationsspiel spie aus zwei Anti-Koordin

ESSs ((x=0,y=1) oder (x=1,y=0) zu denen sich die Populationsgruppen im Laufe der Zeit er stabilen Strategien erreicht wird, hängt von der Anfangsstrategiewahl der Population ab. Die zeitliche Entwicklung von drei Anfangszuständen in einem solchen Eckenspiel. Auch Anfangsstrategiewahl kann es geschehen, dass sich die Population im Laufe der Zeit zu unter grüne und schwarze Trajektorien). Eine besondere Bedeutung hat der Sattelpunkt des Spiels (Sattelpunktes im x-y Diagramm entspricht dem Wert des gemischten Nash-Gleichgewichtes der Funktionen $g_A(x, y)$ und $g_B(x, y)$ bestimmen. Bei einem Zentrumsspielen existiert ke Population sich im Laufe der Zeit ständig verändert und um ein Zentrum kreist. Die Positior entspricht dem Wert des gemischten Nash-Gleichgewichtes bzw. lässt sich durch die Nulls: $g_B(x, y)$ bestimmen (siehe [Evolutionäre Spieltheorie unsymmetrischer](#)

Weiterführende Links

- [Folien der 4. Vorlesung](#)
- [Vorlesungsaufzeichnung der 4. Vorlesung: WS 2022/23 bzw. 1](#)
- [View Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)
- [Download Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)
- [Download Python Programm: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele \(Version 1, Version 2\)](#)
- [View Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)
- [Download Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)
- [Download Python Programm: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele \(Version 1, Version 2\)](#)

Physik der sozio-ökonomischen Systeme mit dem Computer

(Physics of Socio-Economic Systems with the Computer)

Vorlesung gehalten an der J.W.Goethe-Universität in Frankfurt am Main

(Sommersemester 2024)

von Dr.phil.nat. Dr.rer.pol. Matthias Hanauske

Frankfurt am Main 01.02.2024

Erster Vorlesungsteil:

Klassifizierung evolutionärer Bi-Matrix Spiele (unsymmetrische (2 × 2)-Spiele)

Einführung

In diesem Unterkapitel werden die unterschiedlichen Spieltypen evolutionärer Bi-Matrix Spiele (unsymmetrische (2 × 2)-Spiele) klassifiziert. Ausgangspunkt sind die folgenden allgemeine

unsymmetrische Auszahlungsmatrizen nehmen wir das Folgende an: $\hat{\$}^B \neq (\hat{\$}^A)^T$.

$$\hat{\$}^A = \begin{pmatrix} \$_{11}^A & \$_{12}^A \\ \$_{21}^A & \$_{22}^A \end{pmatrix}, \quad \hat{\$}^B = \begin{pmatrix} \$_{11}^B & \$_{12}^B \\ \$_{21}^B & \$_{22}^B \end{pmatrix}$$

Unsymmetrische (2 × 2) Spiele lassen sich in die folgenden Spielklassen gliedern:

Die Klasse der Eckenspiele (engl.: corner class games)

Ein Eckenspiel liegt vor, falls eine der Auszahlungsmatrizen der Spielergruppen eine dominante Struktur hat; falls $\A oder $\B ein dominantes Spiel ist.

Die Klasse der Sattelpunktspiele (engl.: saddle class games)

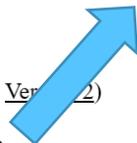
Ein Sattelpunktspiel liegt vor, falls beide Spielergruppen gleichzeitig ein Koordinationsspiel oder Anti-Koordinationsspiel spielen.

Die Klasse der Zentrumsspiele (engl.: center class games)

Ein Zentrumsspiel liegt vor, falls Spielergruppe A ein Koordinationsspiel und Spielergruppe B ein Anti-Koordinationsspiel spielen; falls Spielergruppe A ein Anti-Koordinationsspiel und Spie

Die beiden Komponenten der zweidimensionalen gruppenspezifischen Populationsvektoren lassen sich, aufgrund ihrer Normalisierungsbedingung, auf eine Komponente reduzieren (x_2^A Populationsvektoren (Gruppe A: $x(t) := x_1^A(t)$ und Gruppe B: $y(t) := x_1^B(t)$) wird in der Reproduktionsdynamik mittels des folgenden Systems von Differenzialgleichungen beschrieben:

$$\frac{dx(t)}{dt} = [(\$_{11}^A + \$_{22}^A - \$_{12}^A - \$_{21}^A) y(t) + (\$_{12}^A - \$_{22}^A)] (x(t) - (x(t))^2) =: g_A(x, y) \quad (2)$$



Jupyter Notebook

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import matplotlib
from scipy.integrate import solve_ivp
```

Definition der Funktion $g_A(x, y)$ und $g_B(x, y)$.

```
In [2]: def gA(x, y, Aa, Ab, Ac, Ad):
    g = ((Aa+Ad-Ac-Ab)*y + (Ab-Ad))*(x-x*x)
    return g
def gB(x, y, Ba, Bb, Bc, Bd):
    g = ((Ba+Bd-Bc-Bb)*x + (Bb-Bd))*(y-y*y)
    return g
```

Definition des Systems der gekoppelten Differenzialgleichung des evolutionären Bi-Matrix Spiels.

```
In [3]: def DGLsys(t, vx):
    x, y = vx
    dxdt = gA(x, y, Aa, Ab, Ac, Ad)
    dydt = gB(x, y, Ba, Bb, Bc, Bd)
    return [dxdt, dydt]
```

Beispiel I eines Eckenspiels

Wir betrachten die zeitliche Entwicklung eines Eckenspiels mit den folgenden Auszahlungswerten der beiden Spielergruppen:

$$\hat{\$}^A = \begin{pmatrix} 10 & 4 \\ 12 & 5 \end{pmatrix}, \quad \hat{\$}^B = \begin{pmatrix} 10 & 7 \\ 12 & 5 \end{pmatrix}$$

Es handelt sich hierbei um eine Kombination von einem dominanten Spiel (Spielergruppe A: dominante Strategie 2 ($x=0$)) mit einem Anti-Koordinationsspiel (Spielergruppe B: zwei unsymmetrische Nash-Gleichgewichte).

Im Folgenden wird das evolutionäre Eckenspiel für die Anfangsbedingung $(x_0, y_0) = (0.9, 0.2)$ berechnet:

```
In [4]: Aa, Ab, Ac, Ad = 10, 4, 12, 5
Ba, Bb, Bc, Bd = 10, 7, 12, 5
t_val = np.linspace(0, 7, 1000)
x0 = 0.9
y0 = 0.2
Loes = solve_ivp(DGLsys, [0, 7], [x0, y0], t_eval=t_val)
```

Weiterführende Links

Folien der 4. Vorlesung

Vorlesungsaufzeichnung der 4. Vorlesung: [WS 2022/23](#) bzw. [WS 2021/22](#)

[View Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)

[Download Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)

Download Python Programm: Evolutionäre Spieltheorie symmetrischer (2x2)-Spiele ([Version 1](#) , [Version 2](#))

[View Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)

[Download Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)

Download Python Programm: Evolutionäre Spieltheorie unsymmetrischer (2x2)-Spiele ([Version 1](#) , [Version 2](#))

-
-
-
-
-
-
-
-

Auf der Internetseite der Vorlesung

Python Programm Version 2 (bimatrix1a.py)

Zum Lösen des evolutionären Spiels mit Python werden wir in dieser Version die Funktion '[solve_ivp\(...\)](#)' verwenden, die sich im Untermodul '[scipy.integrate](#)' befindet, welche Funktionen zum Lösen von gewöhnlichen Differenzialgleichungen bereitstellt.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib
4 from scipy.integrate import solve_ivp
5
6 # Definition der Funktionen g_A und g_B
7 def gA(x,y,a,b,c,d):
8     g = ((a+d-c-b)*y + (b-d))*(x-x*x)
9     return g
10
11 def gB(x,y,a,b,c,d):
12     g = ((a+d-c-b)*x + (b-d))*(y-y*y)
13     return g
14
15 # Definition der DGL Systems
16 def DGLsys(t,vx):
17     x, y = vx
18     dxdt = gA(x,y,Aa,Ab,Ac,Ad)
19     dydt = gB(x,y,Ba,Bb,Bc,Bd)
20     return [dxdt,dydt]
21
22 # Groessenfestlegung der Labels usw. im Bild
23 params = {
24     'text.usetex'      : True,
25     'axes.titlesize'   : 22,
26     'axes.labelsize'   : 20,
27     'xtick.labelsize'  : 20 ,
28     'ytick.labelsize'  : 20
29 }
30 matplotlib.rcParams.update(params)
31
32 #Festlegung der Auszahlungsmatrix des unsymmetrischen (2x2)-Spi
33 # Klasse der Eckenspiele (Corner Class Games)
34 #Aa=10
35 #Ab=4
```

Lösen des evolutionären Spiels mit Python (Version 2)

```
64 #Bc=12
65 #Bd=5
66 #tend=8.5
67 #x0=0.6
68 #y0=0.01
69
70 # Weitere Festlegungen
71 numpoints = 1000
72 t_val = np.linspace(0, tend, numpoints)
73
74 # Loesung der DGL fuer eine Anfangspopulation (x0,y0)
75 Loes = solve_ivp(DGLsys, [0, tend], [x0, y0], t_eval=t_val)
76
77 # Fuer die Darstellung des Feldliniendiagramms streamplot
78 SY,SX = np.mgrid[0:1:100j,0:1:100j]
79 SgA = gA(SX,SY,Aa,Ab,Ac,Ad)
80 SgB = gB(SX,SY,Ba,Bb,Bc,Bd)
81 # Die Farbe wird die Geschwindigkeit der Aenderung des Populationsvektors anzeigen
82 speed = np.sqrt(SgA*SgA + SgB*SgB)
83 colorspeed = speed/speed.max()
84
85 # Plotten des Bildes
86 fig, ax = plt.subplots()
87 strm = ax.streamplot(SX,SY,SgA,SgB,density=[2, 2], linewidth=1,color=colorspeed, cmap=plt.cm.cool)
88 ax.plot(Loes.y[0],Loes.y[1],c="black", linewidth=1.5, linestyle='-')
89 ax.plot(x0,y0, marker='o', color='grey', markersize=8)
90 for i in np.linspace(numpoints/30,numpoints-2,10):
91     ax.arrow(Loes.y[0][int(i)],Loes.y[1][int(i)], Loes.y[0][int(i)+1] - Loes.y[0][int(i)], Loes.y[1][int(i)+1] - Loes.y[1][int(i)], head_width=0.02, head_length=0.03,
92             fc='black', ec='black')
93 # Erzeugung der nebenstehenden Farblegende colorbar
94 cbar = fig.colorbar(strm.lines,ax=ax,pad=0.02)
95 cbar.set_label(r'\sqrt{{g_A}^2 + {g_B}^2}',size=20)
96
97 # Achsenbeschriftungen usw.
98 plt.xlim(0,1)
99 plt.ylim(0,1)
100 plt.xticks([0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],["$0$", "", "$0.2$", "", "$0.4$", "", "$0.6$", "", "$0.8$", "", "$1$"])
101 plt.yticks([0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],["$0$", "", "$0.2$", "", "$0.4$", "", "$0.6$", "", "$0.8$", "", "$1$"])
102 plt.ylabel(r"$\rm x$")
103 plt.xlabel(r"$\rm y$")
104
105 #Speichern der Bilder als (.png , benötigt dvi png unter Linux)- und .pdf-Datei
106 saveFig="./bimatrix.png"
107 plt.savefig(saveFig, dpi=100, bbox_inches="tight", pad_inches=0.05, format="png")
108 saveFig="./bimatrix.pdf"
109 plt.savefig(saveFig, bbox_inches="tight", pad_inches=0.05, format="pdf")
110 plt.show()
```

Bi-Matrix Spiele mit Python
(Feldliniendiagramm)

Bi-Matrix Spiele mit Python V2 (Feldliniendiagramm)

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib

# Definition der Funktionen g_A und g_B
def gA(x,y,a,b,c,d):
    g=((a+d-c-b)*y + (b-d))*(x-x*x)
    return g
def gB(x,y,a,b,c,d):
    g=((a+d-c-b)*x + (b-d))*(y-y*y)
    return g

# Groessenfestlegung der Labels usw. im Bild
params = {
    'text.usetex'      : True,
    'axes.titlesize'  : 22,
    'axes.labelsize'  : 20,
    'xtick.labelsize' : 20 ,
    'ytick.labelsize' : 20
}
matplotlib.rcParams.update(params)

#Festlegung der Auszahlungsmatrix des
Aa=10
Ab=4
Ac=9
Ad=5
Ba=10
Bb=4
Bc=7
Bd=5
tend=12
x0=0.6
y0=0.1
```

```
# Fuer die Darstellung des Feldliniendiagramms streamplot
SY,SX=np.mgrid[0:1:100j,0:1:100j]
SgA=gA(SX,SY,Aa,Ab,Ac,Ad)
SgB=gB(SX,SY,Ba,Bb,Bc,Bd)
# Die Farbe wird die Geschwindigkeit der Aenderung des Populationsvektors anzeigen
speed=np.sqrt(SgA*SgA + SgB*SgB)
colourspeed = speed/speed.max()

# Plotten des Bildes
strm = plt.streamplot(SX,SY,SgA,SgB,density=[2, 2], linewidth=1,color=colourspeed, cmap=plt.cm.cool)
# Erzeugung der nebenstehenden Farblegende colorbar
cbar=plt.colorbar(strm.lines,pad=0.02)
cbar.set_label(r'$\sqrt{\{g_A\}^2 + \{g_B\}^2}$',size=20)

# Achsenbeschriftungen usw.
plt.xlim(0,1)
plt.ylim(0,1)
plt.ylabel(r"$\rm y$")
plt.xlabel(r"$\rm x$")

#Speichern der Bilder als .jpg- und .pdf-Datei
saveFig="./bimatrix.jpg"
plt.savefig(saveFig, dpi=100,bbox_inches="tight",pad_inches=0.05,format="jpg")
plt.show()
```

$g_B(x, y)$ bestimmen (siehe [Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)).

Weiterführende Links

- [Folien der 4. Vorlesung](#)
- Vorlesungsaufzeichnung der 4. Vorlesung: [WS 2022/23](#) bzw. [WS 2021/22](#)
- [View Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)
- [Download Jupyter Notebook: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele](#)
- [Download Python Programm: Evolutionäre Spieltheorie symmetrischer \(2x2\)-Spiele \(Version 1, Version 2\)](#)
- [View Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)
- [Download Jupyter Notebook: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele](#)
- [Download Python Programm: Evolutionäre Spieltheorie unsymmetrischer \(2x2\)-Spiele \(Version 1, Version 2\)](#)

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib
4
5 # Definition der Funktionen g_A und g_B
6 def gA(x,y,a,b,c,d):
7     g = ((a+d-c-b)*y + (b-d))*(x-x*x)
8     return g
9
10 def gB(x,y,a,b,c,d):
11     g = ((a+d-c-b)*x + (b-d))*(y-y*y)
12     return g
13
14 # Loesen der DGL
15 def solve_dgl(numpoints,tend,x0,y0,Aa,Ab,Ac,Ad,Ba,Bb,Bc,Bd):
16     sol = np.empty([numpoints,3])
17     t = np.linspace(0,tend,numpoints)
18     dt = t[1] - t[0]
19     sol[0][0] = t[0]
20     sol[0][1] = x0
21     sol[0][2] = y0
22     for i in range(1,numpoints):
23         sol[i][0] = t[i]
24         dx = gA(sol[i-1,1],sol[i-1,2],Aa,Ab,Ac,Ad)*dt
25         dy = gB(sol[i-1,1],sol[i-1,2],Ba,Bb,Bc,Bd)*dt
26         sol[i][1] = sol[i-1,1] + dx
27         sol[i][2] = sol[i-1,2] + dy
28     return sol
29
30 # Groessenfestlegung der Labels usw. im Bild
31 params = {
32     'text.usetex' : True,
33     'axes.titlesize' : 22,
34     'axes.labelsize' : 20,
35     'xtick.labelsize' : 20 ,
36     'ytick.labelsize' : 20
37 }
38 matplotlib.rcParams.update(params)
39
40 #Festlegung der Auszahlungsmatrix des unsymmetrischen (2x2)-Spiels, A
41 # Klasse der Eckenspiele (Corner Class Games)
42 #Aa=10
```

Auf der Internetseite der Vorlesung

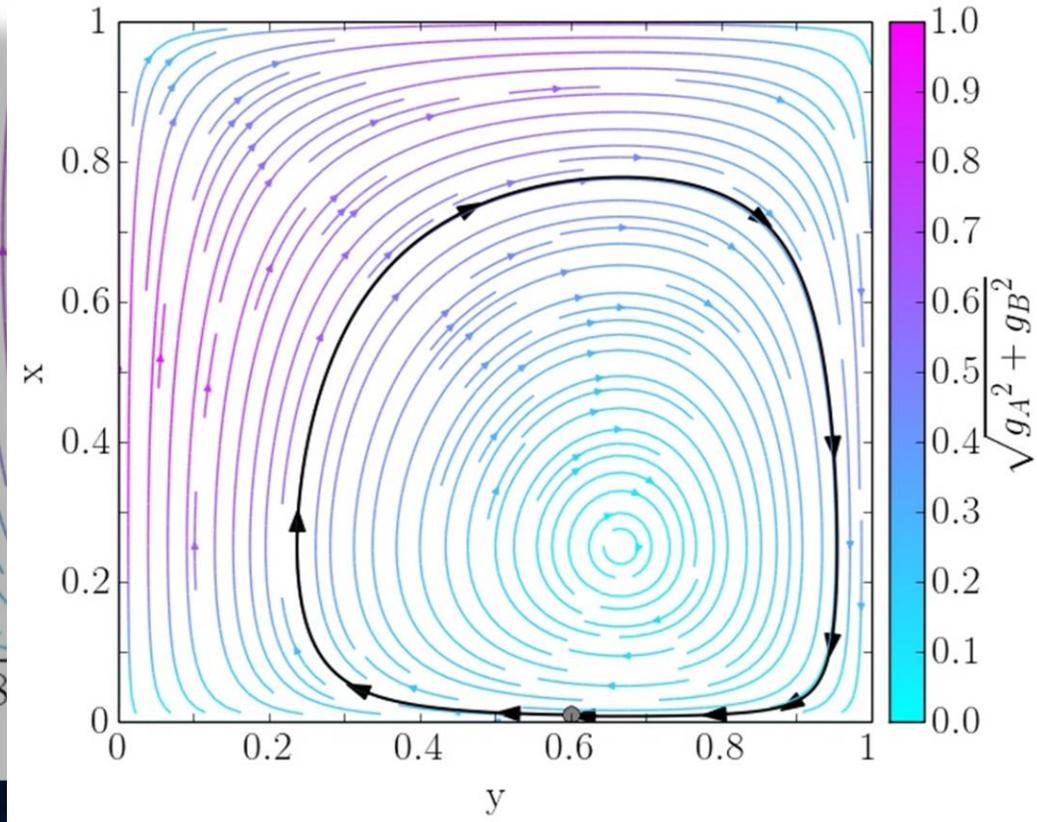
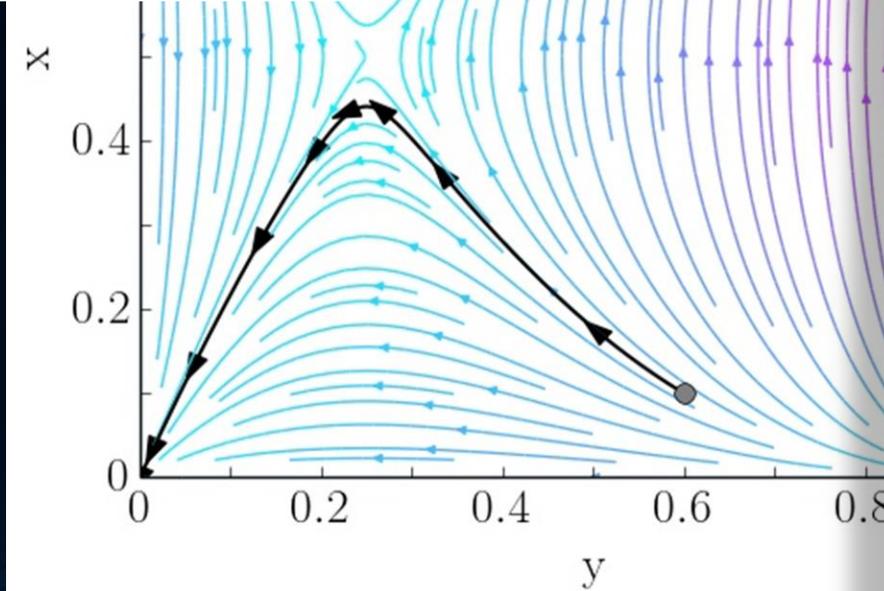
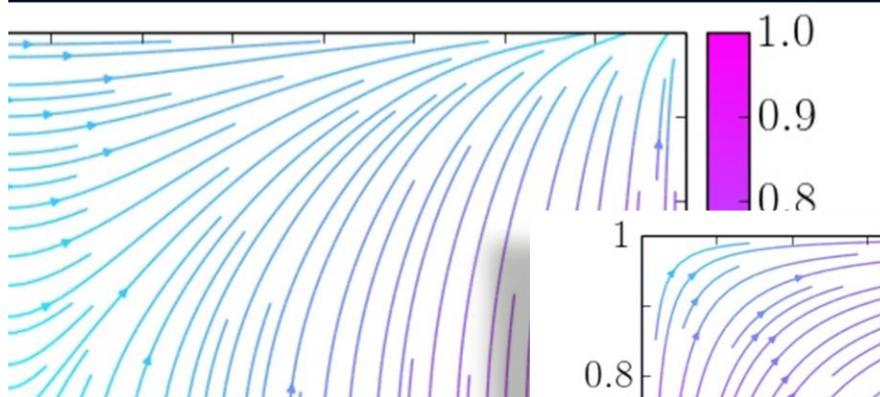
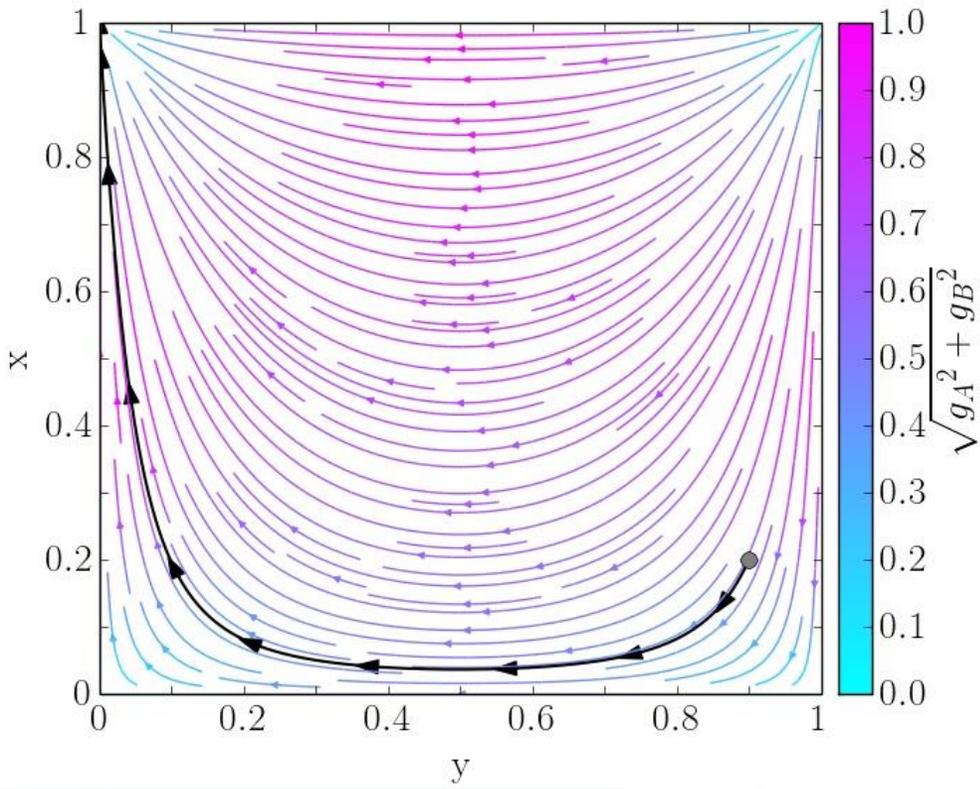
Python Programm
Version 1 (bimatrix1a.py)

Lösen des evolutionären Spiels mit Python,
hier mit dem einfachen Euler-Verfahren

Lösen des evolutionären Spiels mit Python (Version 1)

```
65 # Klasse der Zentrumsspiele (Center Class Games)
66 #Aa=10
67 #Ab=4
68 #Ac=7
69 #Ad=5
70 #Ba=10
71 #Bb=9
72 #Bc=12
73 #Bd=5
74 #tend=8.5
75 #x0=0.6
76 #y0=0.01
77
78 # Weitere Festlegungen
79 numpoints = 1000
80
81 # Loesung der DGL fuer eine Anfangspopulation (x0,y0)
82 Loes = solve_dgl(numpoints,tend,x0,y0,Aa,Ab,Ac,Ad,Ba,Bb,Bc,Bd)
83
84 # Fuer die Darstellung des Feldliniendiagramms streamplot
85 SY,SX = np.mgrid[0:1:100j,0:1:100j]
86 SgA = gA(SX,SY,Aa,Ab,Ac,Ad)
87 SgB = gB(SX,SY,Ba,Bb,Bc,Bd)
88 # Die Farbe wird die Geschwindigkeit der Aenderung des Populationsvektors anzeigen
89 speed = np.sqrt(SgA*SgA + SgB*SgB)
90 colorspeed = speed/speed.max()
91
92 # Plotten des Bildes
93 fig, ax = plt.subplots()
94 strm = ax.streamplot(SX,SY,SgA,SgB,density=[2, 2], linewidth=1,color=colorspeed, cmap=plt.cm.cool)
95 ax.plot(Loes[:,1],Loes[:,2],c="black", linewidth=1.5, linestyle='-')
96 ax.plot(Loes[0,1],Loes[0,2], marker='o', color='grey', markersize=8)
97 for i in np.linspace(numpoints/30,numpoints-2,10):
98     ax.arrow(Loes[int(i),1],Loes[int(i),2], Loes[int(i)+1,1] -Loes[int(i),1], Loes[int(i)+1,2] -Loes[int(i),2], head_width=0.02, head_length=0.03, fc='black', ec='black')
99 # Erzeugung der nebenstehenden Farblegende colorbar
100 cbar = fig.colorbar(strm.lines,ax=ax,pad=0.02)
101 cbar.set_label(r'\sqrt{{g_A}^2 + {g_B}^2}',size=20)
102
103 # Achsenbeschriftungen usw.
104 plt.xlim(0,1)
105 plt.ylim(0,1)
106 plt.xticks([0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],["$0$", "", "$0.2$", "", "$0.4$", "", "$0.6$", "", "$0.8$", "", "$1$"])
107 plt.yticks([0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],["$0$", "", "$0.2$", "", "$0.4$", "", "$0.6$", "", "$0.8$", "", "$1$"])
108 plt.ylabel(r"\rm x$")
109 plt.xlabel(r"\rm y$")
110
111 #Speichern der Bilder als (.png , benötigt dvi/png unter Linux)- und .pdf-Datei
112 saveFig="./bimatrix.png"
113 plt.savefig(saveFig, dpi=100, bbox_inches="tight", pad_inches=0.05, format="png")
114 saveFig="./bimatrix.pdf"
115 plt.savefig(saveFig, bbox_inches="tight", pad_inches=0.05, format="pdf")
116 plt.show()
```

Bi-Matrix Spiele mit Python Version 1 und 2



E-Learning und interaktive Übungsaufgaben

Zusätzlich zu den Informationen aus dieser Internetseite finden Sie in diesem Unterpunkt diverse interaktive Übungsaufgaben zu den folgenden Themen:

Aufgabe 1

Reine Nash-Gleichgewichte in einem simultanen (2x2)-Spiel in strategischer Form mit symmetrischer Auszahlungsmatrix

Aufgabe 2

Gemischtes Nash-Gleichgewicht in einem simultanen (2x2)-Spiel in strategischer Form mit symmetrischer Auszahlungsmatrix

Aufgabe 3

Das gemischtes Nash-Gleichgewicht im Hirschjagt-Spiel

Aufgabe 4

Spielklassen von simultanen (2x2)-Spielen in strategischer Form mit symmetrischer Auszahlungsmatrix

Aufgabe 5

Zeitliche Entwicklung des Populationsvektors im evolutionären Spiel

Aufgabe 6

Evolutionär stabile Strategien

Aufgabe 7

Zeitliche Entwicklung des Populationsvektors im evolutionären Bi-Matrix Spiel

Aufgabe 8

Gemischtes Nash-Gleichgewicht in einem simultanen (2x2)-Spiel in strategischer Form mit unsymmetrischer Auszahlungsmatrix

Aufgabe 9

Gemischtes Nash-Gleichgewicht und zeitliche Entwicklung des Populationsvektors in Zentrumsspielen

Aufgabe 10

Zeitliche Entwicklung des Populationsvektors im evolutionären (2x3)-Spiel

Aufgabe 11

Gemischtes Nash-Gleichgewicht im evolutionären (2x3)-Spiel

Aufgabe 12

Mittlere Distanz zwischen zwei Knoten in einem zufälligen Netzwerk

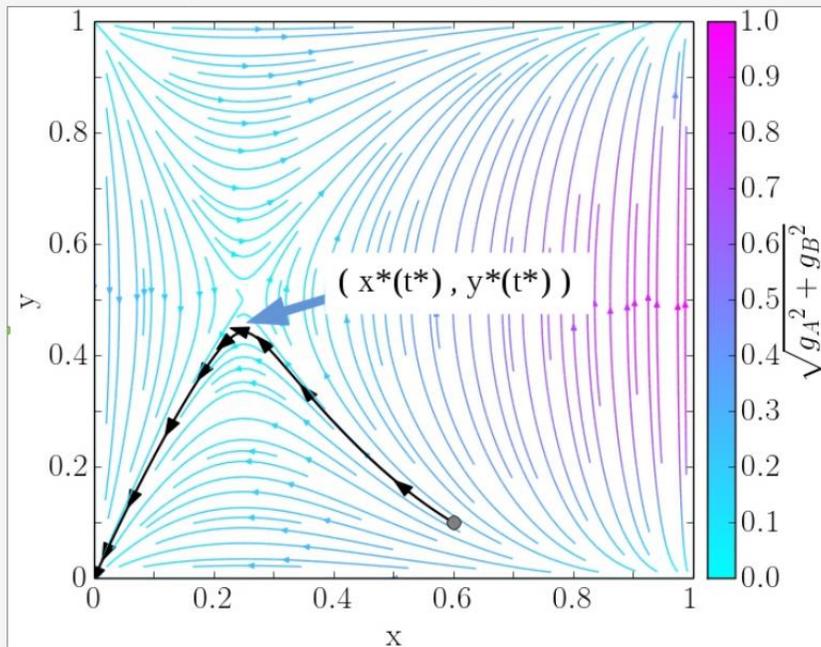
Aufgabe 7

Das zeitliche Verhalten der Komponenten der Populationsvektoren (Gruppe A: $x(t) := x_1^A(t)$ und Gruppe B: $y(t) := x_1^B(t)$) wird in der Reproduktionsdynamik mittels des folgenden Systems von Differentialgleichungen beschrieben:

$$\frac{dx(t)}{dt} = [(\$_{11}^A + \$_{22}^A - \$_{12}^A - \$_{21}^A) y(t) + (\$_{12}^A - \$_{22}^A)] (x(t) - (x(t))^2) =: g_A(x, y)$$

$$\frac{dy(t)}{dt} = [(\$_{11}^B + \$_{22}^B - \$_{12}^B - \$_{21}^B) x(t) + (\$_{12}^B - \$_{22}^B)] (y(t) - (y(t))^2) =: g_B(x, y)$$

Qualitative Veranschaulichung der Aufgabenstellung



Lösung

Der maximale Wert y-Wert beträgt $y^* =$
und er wird zum Zeitpunkt $t^* =$ erreicht.

[Weiter zur nächsten Aufgabe...](#)

Das durch die folgende Auszahlungstabelle definierte Bimatrix Spiel gehört der Klasse der Sattelpunktsspiele an.

A/B	s_1	s_2
s_1	(10 , 10)	(4 , 7)
s_2	(9 , 4)	(5 , 5)

Der Populationsvektor zur Zeit $t=0$ sei $(x(0) = 0.6, y(0) = 0.07882)$. Der Anteil der Spieler in der Gruppe B die die Strategie s_1^B spielen nimmt zunächst zu, erreicht dann ein Maximum und nimmt dann wieder ab (siehe nebenstehende Abbildung). Berechnen Sie den Zeitpunkt t^* an dem der maximale Wert y^* erreicht wird. Tragen Sie bitte die beiden Werte in die unteren Eingabefelder ein

$t^* =$, $y^* =$

und vergleichen Sie indem Sie den folgenden *Button* drücken.

Aufgabe 8

A/B	s_1	s_2
s_1	(210 , 292)	(132 , 120)
s_2	(37 , 125)	(392 , 378)

Betrachten Sie die gemischte Erweiterung eines simultanen (2 Spieler)-(2 Strategien) Spiels in strategischer Form mit unsymmetrischer Auszahlungsmatrix. Die Menge der Spieler sei $\mathcal{I} = \{A, B\}$, die Menge der reinen Strategien sei $\mathcal{S}^A = \mathcal{S}^B = \{s_1, s_2\}$ und die Präferenzordnungen der Spieler sei durch die neben stehende Auszahlungstabelle quantifiziert. Die reinen Strategien entsprechen den folgenden gemischten Strategien: $s_1 \hat{=} \tilde{s}^B = \tilde{s}^B = 1$ und $s_2 \hat{=} \tilde{s}^A = \tilde{s}^B = 0$.

Bei welcher gemischten Strategienkombination $(\tilde{s}^{A*}, \tilde{s}^{B*})$ befindet sich das gemischte Nash-Gleichgewicht? Tragen Sie bitte Ihre Werte in die unteren Eingabefelder ein

$\tilde{s}^{A*} =$, $\tilde{s}^{B*} =$

und vergleichen Sie indem Sie den folgenden *Button* drücken.

Lösung

Das gemischte Nash-Gleichgewicht besindet sich bei

$\tilde{s}^{A*} =$

$\tilde{s}^{B*} =$

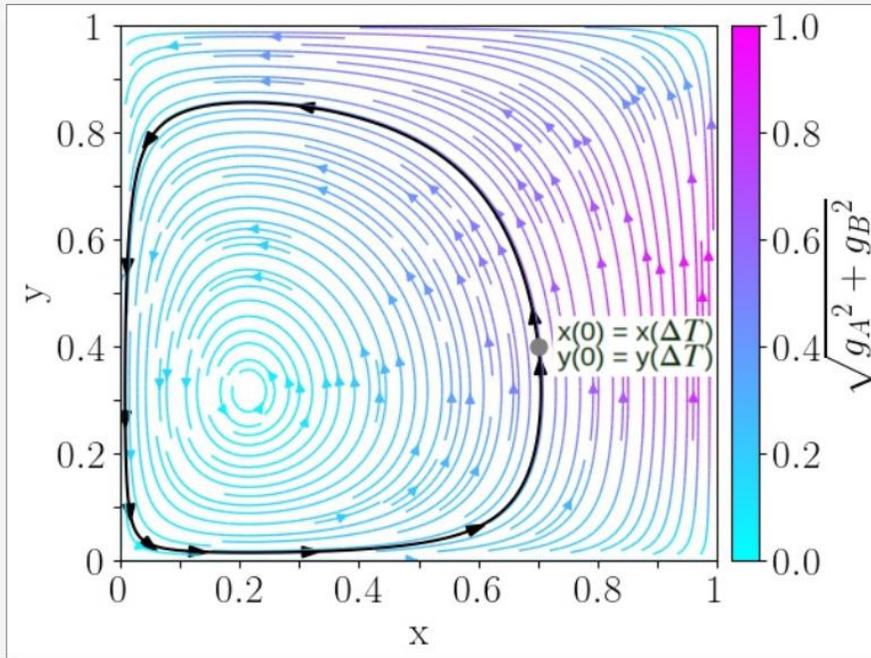
Aufgabe 9

Das zeitliche Verhalten der Komponenten der Populationsvektoren (Gruppe A: $x(t) := x_1^A(t)$ und Gruppe B: $y(t) := x_1^B(t)$) wird in der Reproduktionsdynamik mittels des folgenden Systems von Differentialgleichungen beschrieben:

$$\frac{dx(t)}{dt} = [(\$_{11}^A + \$_{22}^A - \$_{12}^A - \$_{21}^A) y(t) + (\$_{12}^A - \$_{22}^A)] (x(t) - (x(t))^2) =: g_A(x, y)$$

$$\frac{dy(t)}{dt} = [(\$_{11}^B + \$_{22}^B - \$_{12}^B - \$_{21}^B) x(t) + (\$_{12}^B - \$_{22}^B)] (y(t) - (y(t))^2) =: g_B(x, y)$$

Qualitative Veranschaulichung der Aufgabenstellung



Lösung

Die Werte der Auszahlungsmatrix betragen

$c_A =$ und $c_B =$

und die Zeit eines Umlaufes des Populationsvektors beträgt

$\Delta T =$

Das durch die folgende Auszahlungstabelle definierte Bimatrix Spiel gehört der Klasse der Zentrumsspiele an.

A/B	s_1^B	s_2^B
s_1^A	(8 , 8)	(6 , c_B)
s_2^A	(c_A , 4)	(5 , 5)

Der Populationsvektor zur Zeit $t=0$ sei $(x(0)=0.7, y(0)=0.4)$. Wählen Sie die noch offenen Werte der Auszahlungsmatrix (c_A und c_B) so, dass sich das gemischte Nash-Gleichgewicht des Spiels bei $x^* = \bar{s}^{A*} = 0.2741379$ und $y^* = \bar{s}^{B*} = 0.3741379$ befindet. Berechnen Sie dann die Zeit ΔT , die die Population benötigt um wieder zu ihrer Anfangskonfiguration zurück zu kehren: $x(0) = x(\Delta T)$ und $y(0) = y(\Delta T)$. Tragen Sie bitte die berechneten Werte in die unteren Eingabefelder ein

$c_A =$, $c_B =$

$\Delta T =$

und vergleichen Sie indem Sie den folgenden *Button* drücken.