

Physik der sozio-ökonomischen Systeme *mit dem Computer*

*PC-POOL RAUM 01.120
JOHANN WOLFGANG GOETHE UNIVERSITÄT
30.11.2018*

MATTHIAS HANAUSKE

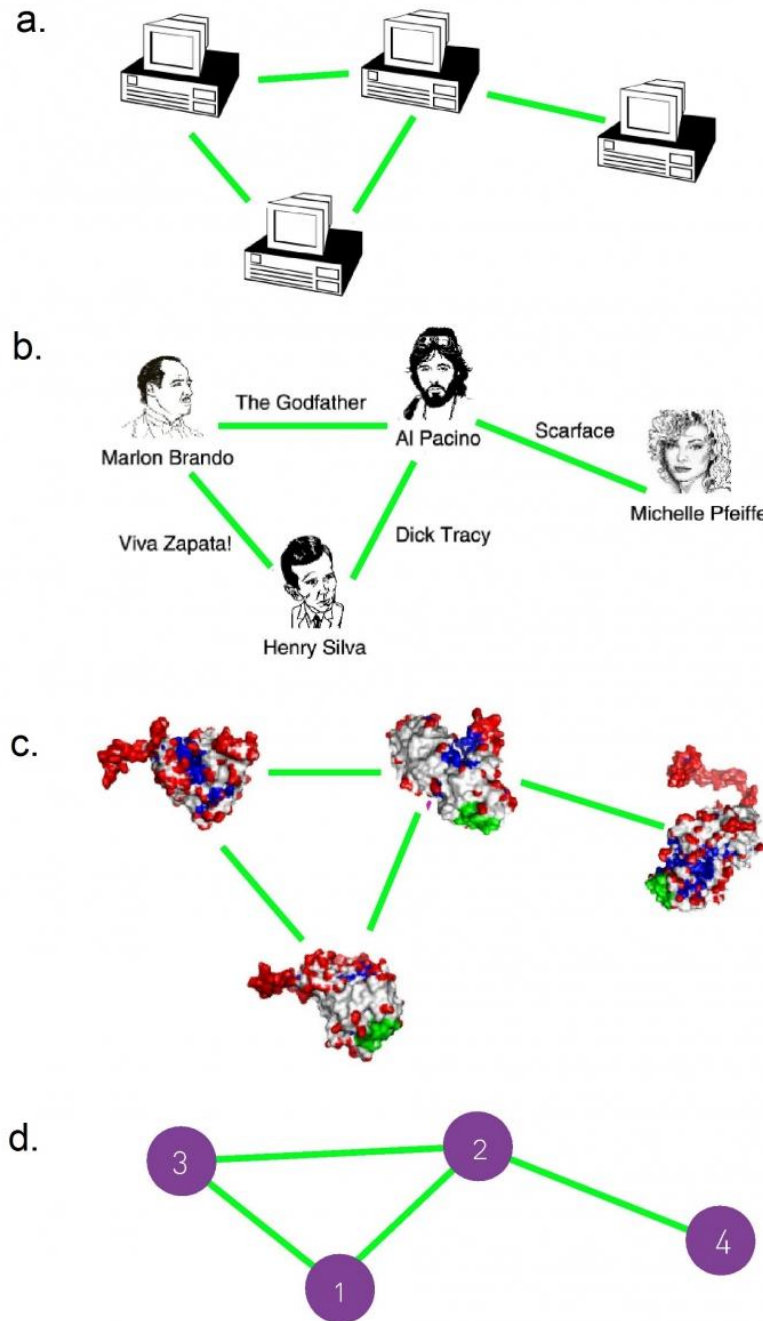
*FRANKFURT INSTITUTE FOR ADVANCED STUDIES
JOHANN WOLFGANG GOETHE UNIVERSITÄT
INSTITUT FÜR THEORETISCHE PHYSIK
ARBEITSGRUPPE RELATIVISTISCHE ASTROPHYSIK
D-60438 FRANKFURT AM MAIN
GERMANY*

7. Vorlesung

Allgemeines zur Vorlesung

- Ort und Zeit:
PC-Pool Raum 01.120, immer freitags von 15.00 bis 17.00 Uhr
- Vorlesungs-Materialien:
<http://th.physik.uni-frankfurt.de/~hанаuske/VPSOC/>
- Aufgaben auf der Online-Lernplattform Lon Capa:
<http://lon-capa.server.uni-frankfurt.de/>
- Plan für die heutige Vorlesung:
Klassen von Netzwerken (zufällige, small world, exponentielle und skalenfreie Netzwerke), Simulation und Darstellung von komplexen Netzwerken mit Python, Simulation und Darstellung von komplexen Netzwerken mit Java

Komplexe Netzwerke Knoten und Kanten



Different Networks, Same Graph

The figure shows a small subset of (a) the Internet, where routers (specialized computers) are connected to each other; (b) the Hollywood actor network, where two actors are connected if they played in the same movie; (c) a protein-protein interaction network, where two proteins are connected if there is experimental evidence that they can bind to each other in the cell. While the nature of the nodes and the links differs, these networks have the same graph representation, consisting of $N = 4$ nodes and $L = 4$ links, shown in (d).

The Structure and Function of Complex Networks*

M. E. J. Newman[†]

Netzwerkstrukturen in unterschiedlichsten Systemen

	Network	Type	n	m
Social	film actors	undirected	449 913	25 516 482
	company directors	undirected	7 673	55 392
	math coauthorship	undirected	253 339	496 489
	physics coauthorship	undirected	52 909	245 300
	biology coauthorship	undirected	1 520 251	11 803 064
	telephone call graph	undirected	47 000 000	80 000 000
	email messages	directed	59 912	86 300
	email address books	directed	16 881	57 029
	student relationships	undirected	573	477
	sexual contacts	undirected	2 810	
Information	WWW nd.edu	directed	269 504	1 497 135
	WWW Altavista	directed	203 549 046	2 130 000 000
	citation network	directed	783 339	6 716 198
	Roget's Thesaurus	directed	1 022	5 103
	word co-occurrence	undirected	460 902	17 000 000
Technological	Internet	undirected	10 697	31 992
	power grid	undirected	4 941	6 594
	train routes	undirected	587	19 603
	software packages	directed	1 439	1 723
	software classes	directed	1 377	2 213
	electronic circuits	undirected	24 097	53 248
	peer-to-peer network	undirected	880	1 296
Biological	metabolic network	undirected	765	3 686
	protein interactions	undirected	2 115	2 240
	marine food web	directed	135	598
	freshwater food web	directed	92	997
	neural network	directed	307	2 359

Netzwerk-Klassen

Aufgrund ihrer unterschiedlichen Eigenschaften unterscheidet man die folgenden Netzwerk-Klassen:

i. Zufällige Netzwerke

Die einzelnen Kanten bei *zufälligen Netzwerke* werden von den Knoten (Spielern) nach einem rein zufälligen Muster ausgewählt.

ii. „Kleine Welt“-Netzwerke (small-world networks)

- i. „Kleine Welt“-Netzwerke zeichnen sich durch einen kleinen Wert der durchschnittlichen kürzesten Verbindung zwischen den Knoten des Netzwerkes und einem großen Wert des Clusterkoeffizienten aus.

iii. Exponentielle Netzwerke

iv. Skalenfreie Netzwerke

Wichtige Netzwerk charakterisierende Größen

Im folgenden werden einige der wichtigen Größen die ein Netzwerk charakterisieren aufgelistet. Eine ausführliche Darstellung der wichtigsten graphentheoretischen Größen findet man unter Albert-Laszlo Barabasi, Network Science, Chapter 2 Graph Theory.

- **Der Knotengrad k_i**

Der Knotengrad des Knotens i ist gleich der Anzahl der Kanten die der Knoten i besitzt. Bei gerichteten Netzwerken unterscheidet man zwischen dem eingehenden k_i^{in} und ausgehenden Knotengrad k_i^{out} . Bei gewichteten Netzwerken summiert man über die Zahlenfaktoren der gewichteten Kanten.

- **Die Nachbarschaftsmatrix (Adjazenzmatrix) A_{ij}**

Die Nachbarschaftsmatrix des Netzwerkes beschreibt, welche Knoten des Graphen durch eine Kante verbunden sind. Zeigt eine Kante von Knoten j auf den Knoten i , dann ist der zugehörige Matrixwert $A_{ij} = 1$, wohingegen ein Eintrag $A_{ij} = 0$ bedeutet, dass keine Kante von j nach i zeigt. Bei ungerichteten Netzwerken ist die Nachbarschaftsmatrix symmetrisch. Der Knotengrad k_i des Knotens i (bzw. k_i^{in} und k_i^{out} bei gerichteten Graphen) lässt sich mittels der Nachbarschaftsmatrix wie folgt bestimmen: $k_i = \sum_{j=1}^N A_{ij}$ ($k_i^{\text{in}} = \sum_{j=1}^N A_{ij}$ und $k_i^{\text{out}} = \sum_{j=1}^N A_{ji}$ bei gerichteten Graphen), wobei die Anzahl der Kanten sich wie folgt bestimmen lässt: $L = \sum_{i,j=1}^N A_{ij}/2$ und die Größe N die Anzahl der Knoten des Netzwerkes bezeichnet.

Wichtige Netzwerk charakterisierende Größen

- **Die Verteilungsfunktion der Knotengrade** $P(k) = N(k)/N$

Die Verteilungsfunktion der Knotengrade $P(k)$ (bzw. $N(k)$) ist eine wichtige das Netzwerk charakterisierende Größe. Sie gibt an, wie groß der Anteil an Netzwerkknoten mit Knotengrad k ist. Bei realen (endlichen) Netzwerken ist diese Funktion keine kontinuierliche, sondern eine diskrete Funktion $P(k) \approx P_k$.

- **Der Clusterkoeffizient** C_i bzw. C

Der Clusterkoeffizient gibt die Wahrscheinlichkeit an, dass zwei nächste Nachbarn eines Knotens ebenfalls nächste Nachbarn untereinander sind. Für einen speziellen Knoten i berechnet er sich mittels:

$C_i = 2E_i / (k_i(k_i - 1))$, wobei E_i die Anzahl der Kanten darstellt, die die nächsten Nachbarn des Knoten i miteinander verbinden. Der globale Wert C des Clusterkoeffizienten ist der Mittelwert aller C_i 's und stellt demnach eine Art von der Enge der Nachbarschaftsbeziehungen des Netzwerks dar.

- **Durchschnittliche Anzahl der m -nächsten Nachbarn** z_m

z_1 stellt hierbei den Wert der mittleren Knotenzahl des Netzwerkes dar ($z_1 = \langle k \rangle$) und z_2 die mittlere Anzahl zweiter-nächster Nachbarn ($z_2 = \langle k^2 \rangle - \langle k \rangle$), siehe z.B. Claudius Gros, "Complex and Adaptive Dynamical Systems, a Primer", S:18).

Wichtige Netzwerk charakterisierende Größen

- **Durchmesser des Netzwerks l**

Der Durchmesser des Netzwerks gibt die maximale kürzeste Kantenlänge zwischen zwei beliebigen Knoten des Netzwerkes an: $l = \log(N/z_1)/\log(z_2/z_1) + 1$, wobei N die Anzahl der Knoten des Netzwerkes darstellt (siehe z.B. Claudius Gros, "Complex and Adaptive Dynamical Systems, a Primer", S:20). Der Wert l wird auch in einigen Lehrbüchern als d_{\max} bezeichnet (siehe Albert-Laszlo Barabasi, Network science, Chapter 2 Graph Theory).

- **Größter verbundener Knotenclusters N_G (Giant component, Hub)**

Die Anzahl der Knoten im größten verbundenen Knotenclusters des Netzwerkes wird mit N_G bezeichnet.

Diese und weitere graphentheoretischen Größen (Number of Shortest Paths Between Two Nodes, Shortest Path, Average Path Length, ...) sind in Albert-Laszlo Barabasi, Network science, Chapter 2 Graph Theory zu finden.

Aufgrund ihrer unterschiedlichen Eigenschaften unterscheidet man die folgenden Netzwerk-Klassen: Zufällige Netzwerke (Random Networks: Die einzelnen Kanten bei zufälligen Netzwerke werden von den Knoten (Spielern) nach einem rein zufälligen Muster ausgewählt), Kleine Welt-Netzwerke (Small World Networks, Kleine Welt-Netzwerke zeichnen sich durch einen kleinen Wert der durchschnittlichen kürzesten Verbindung zwischen den Knoten des Netzwerkes und einem großen Wert des Clusterkoeffizienten aus), Exponentielle Netzwerke (Exponential Networks) und Skalenfreie Netzwerke (Scale-Free Networks).

Section 3.2

The Random Network Model

Wie erzeugt man mittels eines mathematischen Algorithmus ein zufälliges Netzwerk (siehe Box 3.2)

Section 3.4

Degree Distribution

Wie sieht die Verteilungsfunktion der Knotengrade in zufälligen Netzwerken aus (siehe Image 3.4 Binomial vs. Poisson Degree Distribution)

Section 3.5

Real Networks are Not Poisson

Vergleich: Real existierende Netzwerke \leftrightarrow Zufällige Netzwerke (siehe Image 3.6 Degree Distribution of Real Networks)

Section 3.6

The Evolution of a Random Network

Relativen Größe des Hubs (grösster verbundener Knotencluster) hängt von dem durchschnittlichen Knotengrad des Netzwerkes ab. Definition von unterschiedlichen Regimen in zufälligen Netzwerken (subcritical, supercritical, fully connected) (siehe Image 3.7 Evolution of a Random Network)

Section 3.7

Real Networks are Supercritical

Sind real existierende Netzwerke subcritical, supercritical oder fully connected? (siehe Table 3.1 Are Real Networks Connected? und Image 3.9 Most Real Networks are Supercritical)

Section 3.8

Small Worlds

Definition der kleinen Welt Eigenschaft in komplexen Netzwerken "In the language of network science the small world phenomenon implies that the distance between two randomly chosen nodes in a network is short." Mittlerer Abstand zwischen zwei Knoten im Netzwerk $\langle d \rangle$ bestimmt die Eigenschaft von kleinen Welt Netzwerken (siehe Image 3.10 Six Degree of Separation and Image 3.11 Why are Small Worlds Surprising? und Table 3.2 Six Degrees of Separation)

Section 3.9

Clustering Coefficient

Der Clusterkoeffizient in real existierenden und zufälligen Netzwerken (siehe Image 3.13 Clustering in Real Networks und Box 3.9 Watts-Strogatz Model)

Zufällige Netzwerke

Verteilungsfunktion der Knotengrade $P(k)$

VI. CLASSICAL RANDOM GRAPHS, THE ERDÖS-RÉNYI MODEL

The simplest and most studied network with undirected edges was introduced by Erdős and Rényi (ER model) [77,78]. In this network:

- (i) the total number of vertices, N , is fixed;
- (ii) the probability that two arbitrary vertices are connected equals p .

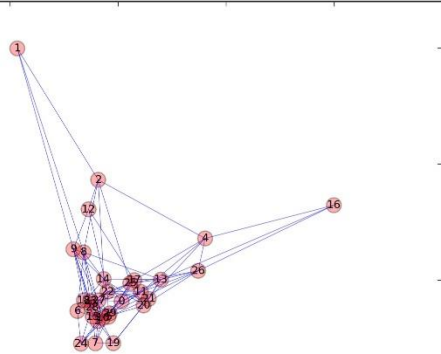
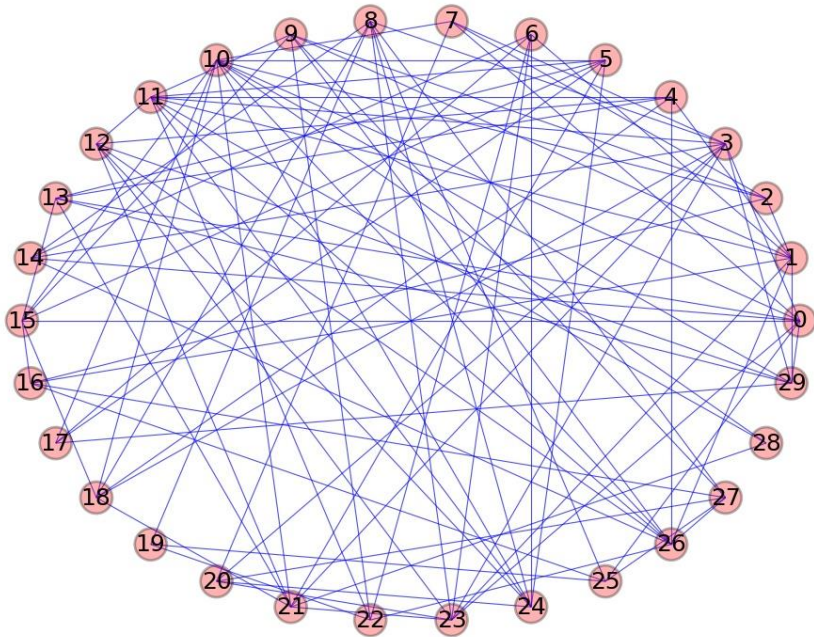
One sees that, on average, the network contains $pN(N-1)/2$ edges. The degree distribution is binomial,

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}, \quad (4)$$

so the average degree is $\bar{k} = p(N-1)$. For large N , the distribution, Eq. (4) takes the Poisson form,

$$P(k) = e^{-\bar{k}} \bar{k}^k / k!. \quad (5)$$

Python (Version 2)



0.5 1.0

```
import networkx as nx
import matplotlib.pyplot as plt
from random import randint
from random import uniform

G=nx.Graph()

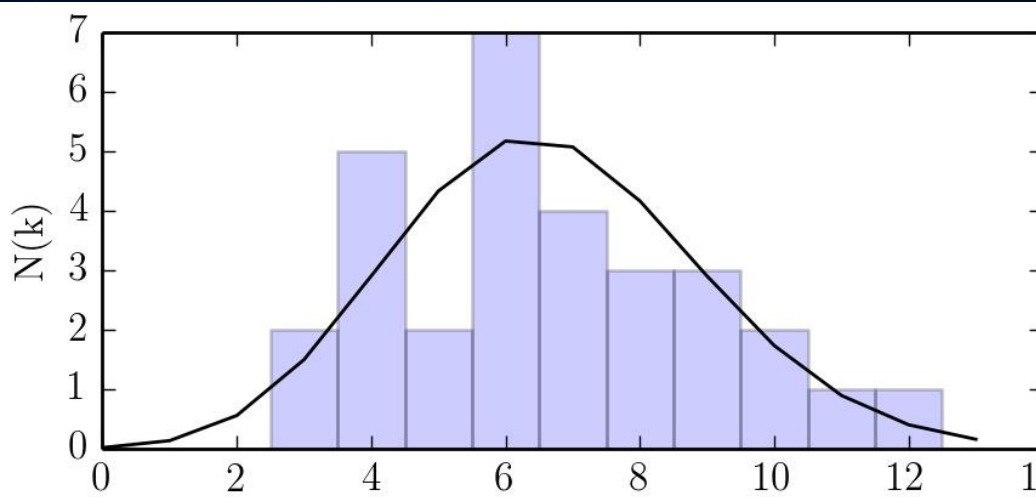
#Allgemeine Festlegungen des Netzwerks
NKn =30 #Anzahl der Knoten (vertices)
anzedges=100 #Gesamte Anzahl der im Netzwerk bestehenden Kanten (links, connections)

#Hinzufuegen der Knoten zum Netzwerk
G.add_nodes_from(range(0,NKn,1))

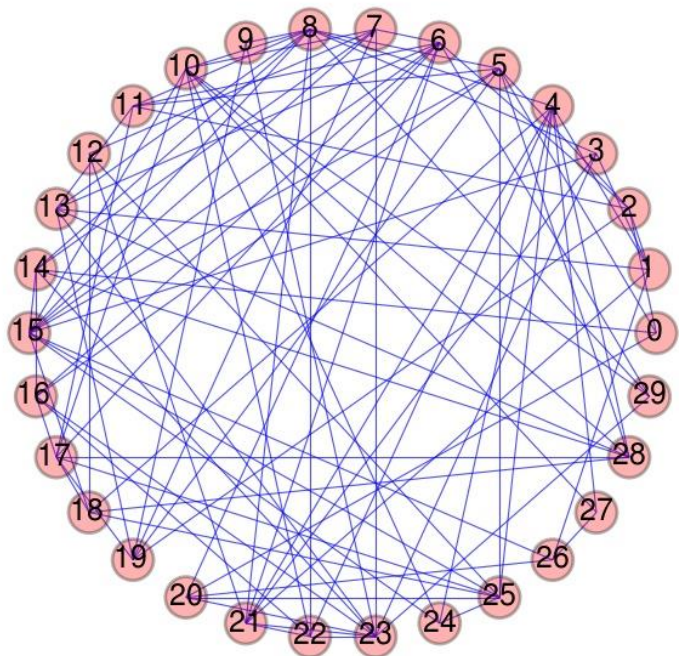
#Erzeugung der Kanten des Netzwerkes (zufaellig anzedges-Kanten zwischen den NKn-Knoten erzeugen)
links=0
while links < anzedges:
    KnA = randint(0, NKn-1)
    KnB = randint(0, NKn-1)
    if KnA != KnB and list(G.edges()).count((KnA,KnB))==0 and list(G.edges()).count((KnB,KnA))==0:
        G.add_edge(KnA,KnB)
        links=links+1

#Erzeugung des Netzwerk-Bildes
node_size=150|
node_alpha=0.3
node_color="red"
edge_tickness=0.4
edge_alpha=0.7
edge_color="blue"
node_text_size=9
text_font="sans-serif"
graph_pos=nx.shell_layout(G)
#graph_pos=nx.spectral_layout(G)
#graph_pos=nx.spring_layout(G)
nx.draw_networkx_nodes(G,graph_pos,node_size=node_size,alpha=node_alpha,node_color=node_color)
nx.draw_networkx_edges(G,graph_pos,width=edge_tickness,alpha=edge_alpha,edge_color=edge_color)
nx.draw_networkx_labels(G,graph_pos,font_size=node_text_size,font_family=text_font)

#Speicherung des Bildes als .jpg und .pdf Datei
saveFig="./Netzwerk.jpg"
```



$$N(k) = N \binom{N-1}{k} p^k (1-p)^{N-1-k}$$



$$N = 30, p = 0.2299, m = 100$$

```

import networkx as nx
import matplotlib.pyplot as plt
from random import randint
from random import uniform
import numpy as np
import matplotlib
import matplotlib.gridspec as gridspec
from matplotlib.ticker import NullFormatter

nullfmt = NullFormatter() # Keine Labels im Netzwerkplot

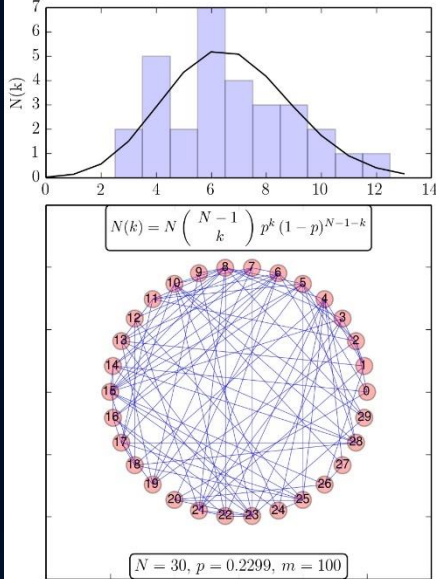
#Von https://de.wikibooks.org/wiki/Algorithmensammlung:_Statistik:_Binomialkoeffizient
def binomialkoeffizient(n, k):
    if k == 0: return 1
    if 2*k > n:
        ergebnis = binomialkoeffizient(n, n-k)
    else:
        ergebnis = n-k+1
        for i in range(2, k+1): # i in [2; k]
            ergebnis *= (n-k+i) # Selbstmultiplikation
            ergebnis /= i # Achtung: Ergebnis ist eine Kommazahl!
    return int(ergebnis)]

#Analytische Verteilungsfunktion eines zufaelligen Netzwerkes
def P(n, kmin, kmax, p):
    ergebnis=[]
    for k in range(kmin, kmax, 1):
        ergebnis.append(binomialkoeffizient(n-1, k)*p**k*(1-p)**(n-1-k))
    return ergebnis

#plot settings
params = {
    'figure.figsize' : [5, 7.2],
    'text.usetex' : True,
}
matplotlib.rcParams.update(params)

#Grid
plt.figure(0)
gs = gridspec.GridSpec(2, 1, height_ratios=[1,2.2], hspace=0.1)
ax1 = plt.subplot(gs[0])
ax2 = plt.subplot(gs[1])

```



```
#####  
#Beginn des eigentlichen Python Programms "Zufaelliches Netzwerk"  
G=nx.Graph()  
#Allgemeine Festlegungen des Netzwerks  
NKn =30 #Anzahl der Knoten (vertices)  
anzedges=100 #Gesamte Anzahl der im Netzwerk bestehenden Kanten (links, connections)  
  
#Hinzufuegen der Knoten zum Netzwerk  
G.add_nodes_from(range(0,NKn,1))  
  
#Erzeugung der Kanten des Netzwerkes (zufaellich anzedges-Kanten zwischen den NKn-Knoten erzeugen)  
links=0  
while links < anzedges:  
    KnA = randint(0, NKn-1)  
    KnB = randint(0, NKn-1)  
    if KnA != KnB and list(G.edges()).count((KnA,KnB))==0 and list(G.edges()).count((KnB,KnA))==0:  
        G.add_edge(KnA,KnB)  
        links=links+1  
  
#Liste der Knotengrade  
degree_sequence=sorted([d for n,d in G.degree()],reverse=True) # degree sequence  
maxk=np.max(degree_sequence)  
  
#Berechnung der Wahrscheinlichkeit das Knoten KnA mit KnB verbunden ist  
p=(2*anzedges/float(NKn*(NKn-1)))  
  
#Erzeugung des Bildes der Verteilungsfunktion der Knotengrade N(k) (analytisch,simulativ)  
ax1.plot(range(0,int(maxk+2),1),np.array(P(NKn,0,int(maxk+2),p))*NKn, linewidth=1, linestyle='-', c="black")  
ax1.hist(degree_sequence,bins=range(0,int(maxk+2),1), align="left", histtype='bar', color="blue", alpha=0.2)  
  
#Achsenbeschriftung  
ax1.set_ylabel(r'$\rm N(k)$')  
ax2.yaxis.set_major_formatter(nullfmt)  
ax2.xaxis.set_major_formatter(nullfmt)  
  
#Erzeugung des Netzwerk-Bildes  
node_size=150  
node_alpha=0.3  
node_color="red"  
edge_tickness=0.4  
edge_alpha=0.7  
edge_color="blue"  
node_text_size=9  
text_font="sans-serif"  
graph_pos=nx.shell_layout(G)  
nx.draw_networkx_nodes(G,graph_pos,node_size=node_size,alpha=node_alpha, node_color=node_color)  
nx.draw_networkx_edges(G,graph_pos,width=edge_tickness, alpha=edge_alpha,edge_color=edge_color)  
nx.draw_networkx_labels(G, graph_pos,font_size=node_text_size,font_family=text_font)  
  
# Plotten der Netzwerkeigenschaften in das Bild  
roundp="%.4f"%p  
textstr1=r'$N(k) = N \left( \begin{array}{c} N-1 \\ k \end{array} \right) p^k (1-p)^{N-1-k}$'  
textstr2=r'$N='+str(NKn)+' , p='+roundp+' , m='+str(anzedges)+'$'  
props = dict(boxstyle='round', facecolor='white', alpha=0.92)  
plt.text(0, 1.46, textstr1, fontsize=11, verticalalignment='top', horizontalalignment='center', bbox=props)  
plt.text(0, -1.46, textstr2, fontsize=12, verticalalignment='bottom', horizontalalignment='center', bbox=props)  
  
#Speicherung des Bildes als .jpg und .pdf Datei  
saveFig="./Netzwerk.jpg"  
plt.savefig(saveFig, dpi=200,bbox_inches="tight",pad_inches=0.05,format="jpg")  
saveFig="./Netzwerk.pdf"  
plt.savefig(saveFig,bbox_inches="tight",pad_inches=0.05,format="pdf")  
plt.show()
```

Python (Version 3)

Konstruktion eines Skalenfreien Netzwerks

Das im folgenden konstruierte skalenfreie Netzwerk besitzt zwei wesentliche Eigenschaften:

- Zeitliches Anwachsen der Knoten
- Die Kantenwahl eines neu in das Netzwerk hinzukommenden Knotens erfolgt nach dem Prinzip des „Preferential Attachment“ (Die Knoten die schon viele Kanten haben bekommen mit einer höheren Wahrscheinlichkeit eine neue Kante, als die Knoten die bisher keinen, oder wenige Kanten aufweisen können)

Exponentielle und Skalenfreie Netzwerke

- Bei exponentiellen und Skalenfreien Netzwerken besitzen viele Knoten wenig Kanten und einige wenige Knoten sehr viele Kanten.
- Im folgenden wollen wir die Konstruktion eines solchen Netzwerks mittels einer Computersimulation betrachten:

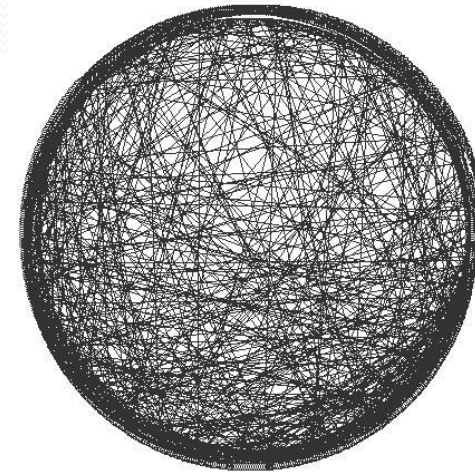
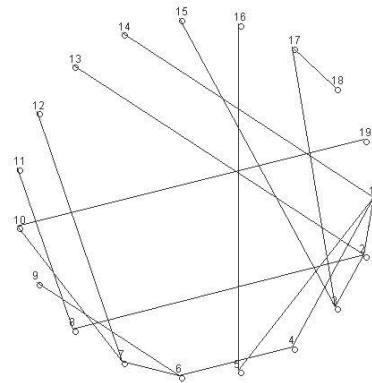
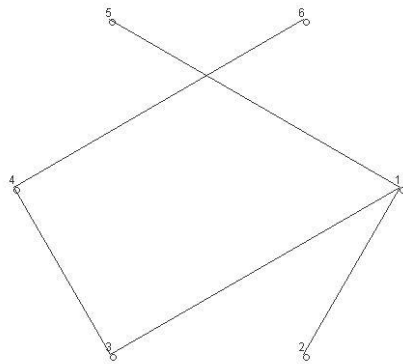


Image 4.2

Die Verteilungsfunktion der Knotengrade des Internets (WWW, links zwischen Internetseiten, gerichtetes Netzwerk) wird mathematisch durch eine „power-law distribution“ beschrieben.

Image 4.4 und 4.5

In skalenfreien Netzwerken entstehen sehr große Hubs (Hauptunterschied zu zufälligen Netzwerken)

Section 4.4:

Was bedeutet „skalenfrei“?

Tabelle 4.1, Image 4.7 und 4.8:

Skalenfreie Netzwerke mit $\gamma < 3$ besitzen keine sinnvolle interne Knotengradskala. Greift man zufällig einen Knoten aus dem Netzwerk, so kann man seinen Knotengrad nicht vorher einschränken; er kann sehr klein oder sehr sehr groß sein. Die Standardabweichung der Knotengrade divergiert sogar für $N \rightarrow \infty$.

Universality Section 4.5, Box 4.2:

Man beobachtet die skalenfreie Eigenschaft in vielen unterschiedlichen real existierenden komplexen Netzwerken

Section 4.6:

Ultra small world Eigenschaft in realen Netzwerken (abhängig vom Exponenten γ)

Box 4.5:

Auswirkung und unterschiedliche Regime in Abhängigkeit vom Exponenten γ .

Box 4.6:

Warum gibt es keine realen Netzwerke mit $\gamma < 2$?

Box 4.9:

Zusammenfassung der mathematischen Eigenschaften von skalenfreie Netzwerken

Das Barabasi-Albert Model der skalenfreien Netzwerke (Section 5)

<http://networksciencebook.com/chapter/5>

Im BA-Model werden die folgenden Prinzipien bei der Konstruktion des skalenfreien Netzwerkes benutzt:

- a) Zeitliches Anwachsen der Knoten und Kanten
- b) Preferential Attachement (Section 5.2, Image 5.2)

Section 5.1:

Konstruktion eines skalenfreien Netzwerkes mit dem Barabasi-Albert Model

Section 5.6:

Vergleich: Exponentielle Netzwerke \leftrightarrow skalenfreie Netzwerke

Box 5.4:

Zusammenfassung der mathematischen Eigenschaften von skalenfreie Netzwerken

Das Java-Applet der Netzwerksimulation

The screenshot shows a Java applet window titled "Applet Netzwerk3 started". The interface includes a network graph on the left, control buttons for "Start" and "Stop", and a table of simulation data. Below the table are six plots arranged in a 2x3 grid, showing the distribution of nodes and their logarithmic transformations for two different parameter sets.

Thread counter t= 645

Start

Stop

Erscheinung:

Anz. Zitate pro Periode (z):

Anz. Papers pro Periode (M):

Anfangszitate (k0):

$N(0) = 336$	$\log(N(0)) = 5.817111159963204$
$N(1) = 150$	$\log(N(1)) = 5.0106352940962555$
$N(2) = 75$	$\log(N(2)) = 4.31748811353631$
$N(3) = 42$	$\log(N(3)) = 3.7376696182832684$
$N(4) = 12$	$\log(N(4)) = 2.4849066497880004$
$N(5) = 18$	$\log(N(5)) = 2.8903717578961645$
$N(6) = 5$	$\log(N(6)) = 1.6094379124341003$
$N(7) = 6$	$\log(N(7)) = 1.791759469228055$
$N(8) = 0$	$\log(N(8)) = -\text{Infinity}$
$N(9) = 1$	$\log(N(9)) = 0.0$
$N(10) = 0$	$\log(N(10)) = -\text{Infinity}$
$N(11) = 0$	$\log(N(11)) = -\text{Infinity}$

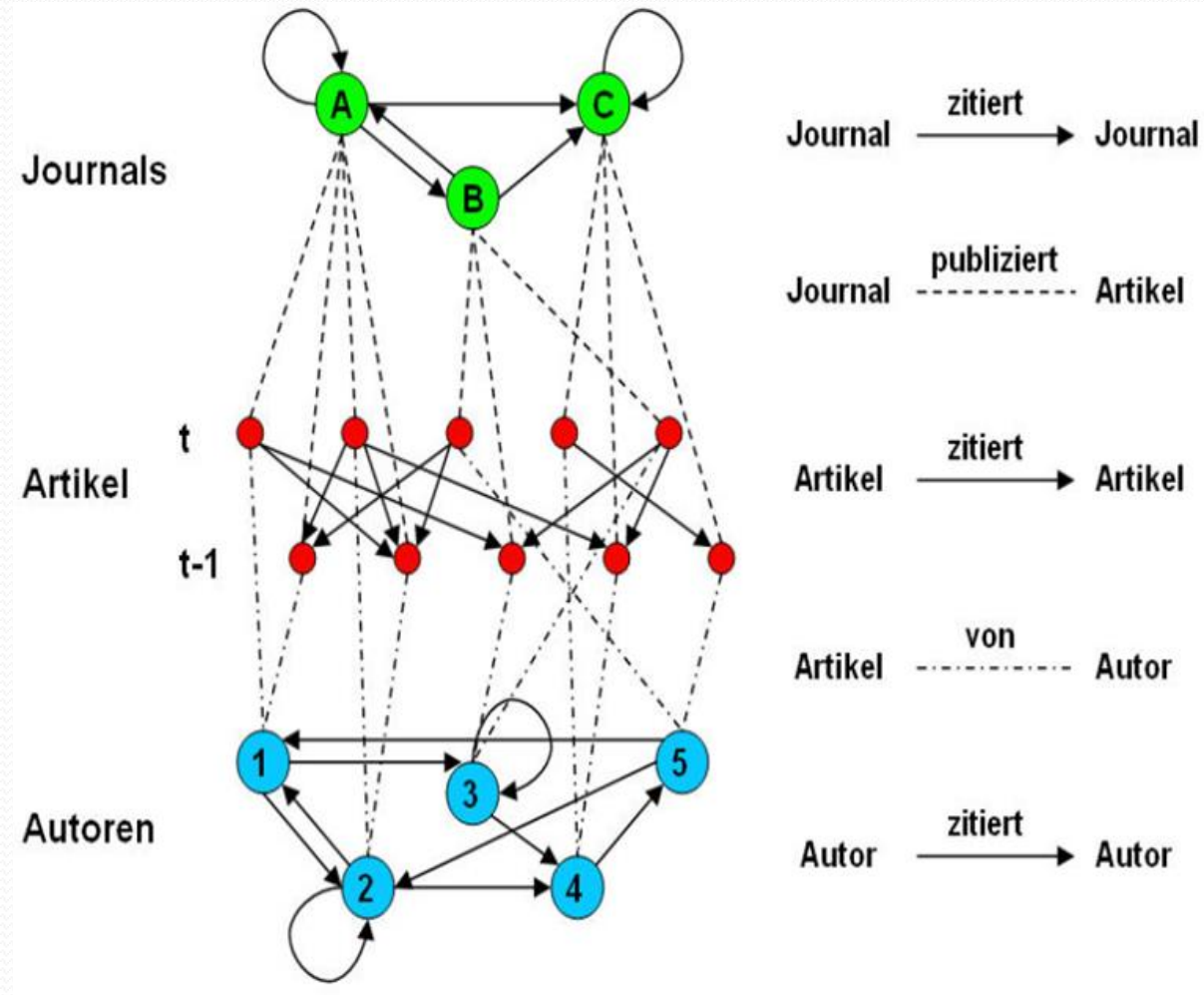
Row 1 plots (Parameters: $z=1, M=1, k_0=1$):

- $N(k)$ plot: x-axis from (0,0,0) to (11,0,0), y-axis from (0,0,336,0) to (11,0,338,0)
- $\ln(N(k))$ plot: x-axis from (0,0,0) to (11,0,0), y-axis from (0,0,5,823) to (11,0,5,823)
- $\ln(N(\ln(k)))$ plot: x-axis from (0,0,0) to (2,398,0,0), y-axis from (0,0,5,823) to (11,0,5,823)

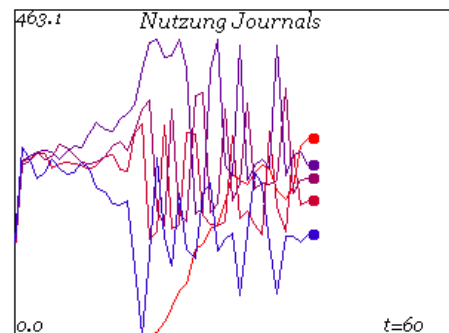
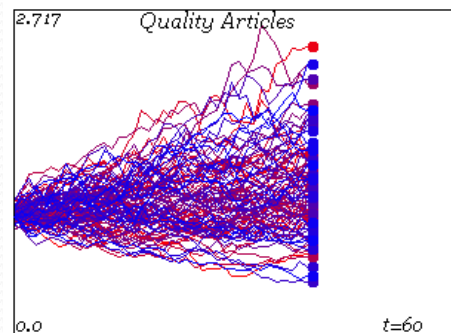
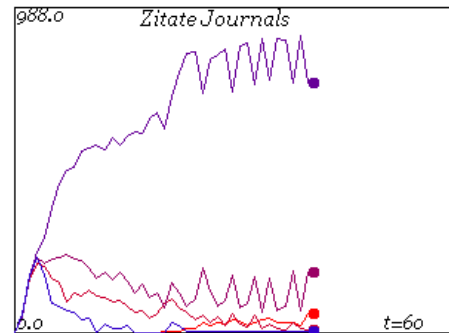
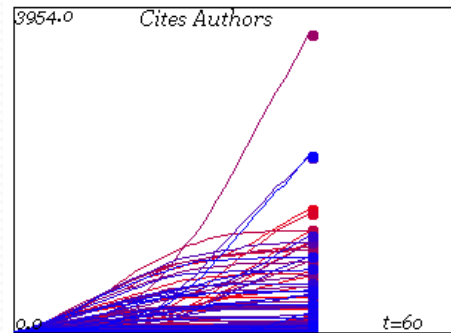
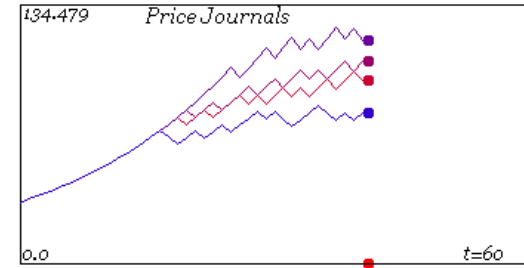
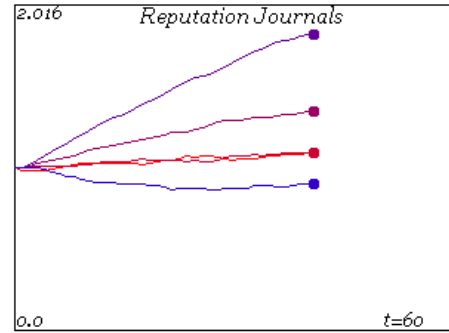
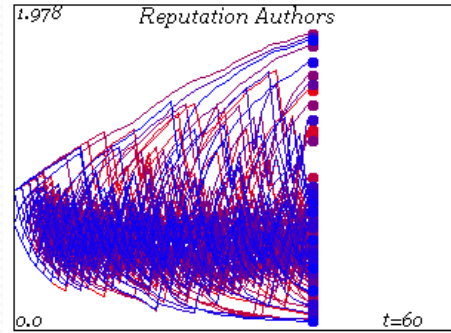
Row 2 plots (Parameters: $z=1, M=1, k_0=1$):

- $N(k)$ plot: x-axis from (0,0,0) to (11,0,0), y-axis from (0,0,432,0) to (11,0,432,0)
- $\ln(N(k))$ plot: x-axis from (0,0,0) to (11,0,0), y-axis from (0,0,6,066) to (11,0,6,066)
- $\ln(N(\ln(k)))$ plot: x-axis from (0,0,0) to (2,398,0,0), y-axis from (0,0,6,066) to (11,0,6,066)

Schematische Darstellung des implementierten Zitationsnetzwerks



Das Java Simulationsapplet

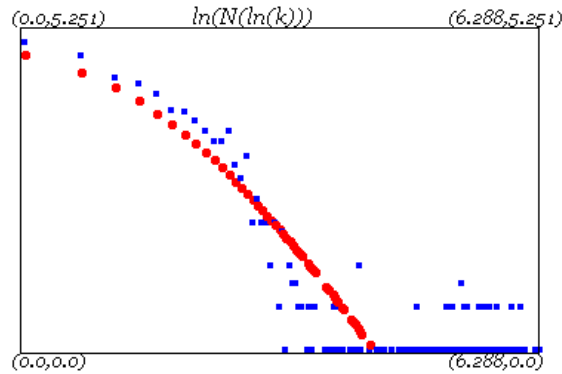


Starting: All Equal Pricing: auch runter

Start Stop

WIAP Wissenschaftliche Informationsversorgung und alternative Preisbildungsmechanismen

Periods	Cites New	FT	sigma	oa Anz
60	2	20	0.05	25
Authors	Cites Old	Read Papers	Bibs	oa Start
100	8	5	3	20
Journals	alpha 1	Pub.Articles	Staat	Lesevers
5	1	12	800	3
Forgetrange	alpha 2	mu	Staatd	
30	1.2	0.005	1	



- **- Die evolutionäre Entwicklung sozio-ökonomischer Systeme**
- **aus dem Blickwinkel der Theoretischen Physik -**
- **Dr. phil. nat. Dr. rer. pol. Matthias Hanauske**

Die innerhalb der soziologischen Forschung entwickelte Akteur-Netzwerk-Theorie (ANT siehe z.B. Bruno Latour 1980) beschreibt die Gesellschaft bzw. die Welt als ein netzwerkartiges Gebilde aus handelnden Akteuren (Aktanten), die in netzwerkartigen Handlungszusammenhängen agieren. Innerhalb der theoretischen Physik entwickelte sich seit 1999 ein mathematisch theoretisches Teilgebiet, das sich mit der Erforschung der Physik sozio-ökonomischer Systeme beschäftigt. Ist es möglich, das Verhalten einer Gruppe von Entscheidungsträgern mathematisch zu beschreiben und die zeitliche Entwicklung ihrer Strategiewahl vorherzusagen? Gibt es eine Art von fundamentaler Theorie des menschlichen Verhaltens, ähnlich der physikalischen Elementarteilchen-Theorie und ihrer fundamentalen Wechselwirkungen?

Im Jahre 1928 veröffentlichte John von Neumann eine mathematische Analyse mit dem Titel "Zur Theorie der Gesellschaftsspiele" und legte damit die Grundlagen der heutigen Spieltheorie, die viele Wirtschafts- und Sozialwissenschaftler als die formale Sprache der ökonomischen Theorie betrachten. Die zeitliche Veränderung des menschlichen Verhaltens, die evolutionäre Dynamik interagierender Menschengruppen unterliegt gewissen mathematischen Prinzipien, nach denen sich das Leben entwickelte und auch in Zukunft weiter entwickeln wird.

Nach einer kurzen Einführung in die evolutionäre Spieltheorie auf sozio-ökonomischen komplexen Netzwerken wird ein neuer Ansatz (die Quanten-Spieltheorie) im Detail vorgestellt. Die Quanten-Spieltheorie stellt eine mathematische und konzeptuelle Erweiterung der klassischen Spieltheorie dar. Der Raum aller denkbaren Entscheidungswege der Akteure wird vom rein reellen, messbaren Raum in den Raum der komplexen Zahlen (reelle und imaginäre Zahlen) ausgedehnt. Durch das Konzept der möglichen quantentheoretischen Verschränkung der Entscheidungswege im imaginären Raum aller denkbaren Quantenstrategien können gemeinsame, durch kulturelle oder moralische Normen entstandene Denkrichtungen in die evolutionäre Dynamik mit einbezogen werden. Ist die Strategiever-schränkung der beteiligten Akteure im imaginären Raum der denkbaren Entscheidungswege nur genügend groß, so können zusätzliche Nash-Gleichgewichte auftreten und sich zuvor existente dominante Strategien mit dilemmaartigen Strukturen auflösen.

Dr. phil. nat. Dr. rer. pol. Matthias Hanauske studierte in Frankfurt und Konstanz Physik und promovierte in Physik 2004 und in Wirtschaftswissenschaften 2011 an der Goethe-Universität in Frankfurt am Main. Z.Zt. ist er am FIAS (Frankfurt Institute of Advanced Studies) aktiv.

Email: info@hinterdernatur.de

www.hinterdernatur.de



Hinter der Natur e.V.

Verein für Philosophie und Naturwissenschaft

Vortrag

Mo, 10. Dezember 2018, 19.30 Uhr

Denkbar, Spohrstraße 46a, Frankfurt

- **- Die evolutionäre Entwicklung sozio-ökonomischer Systeme**

Vergleich des simulierten Artikelnetzwerks mit empirischen Daten

Das auf der Artikelebene simulierte Zitationsnetzwerk (Abbildung b) stimmt gut mit der in Realität beobachteten Netzwerkstruktur (Abbildung a) überein. In Abbildung a sind die Zitationsnetzwerke der Zeitschrift *Physical Review D* und der Datenbank *ISI (Institute of scientific Information)* aufgetragen.

