

# Allgemeine Relativitätstheorie mit dem Computer

*PC-POOL RAUM 01.120  
JOHANN WOLFGANG GOETHE UNIVERSITÄT  
29. MAI, 2020*

*MATTHIAS HANAUSKE*

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

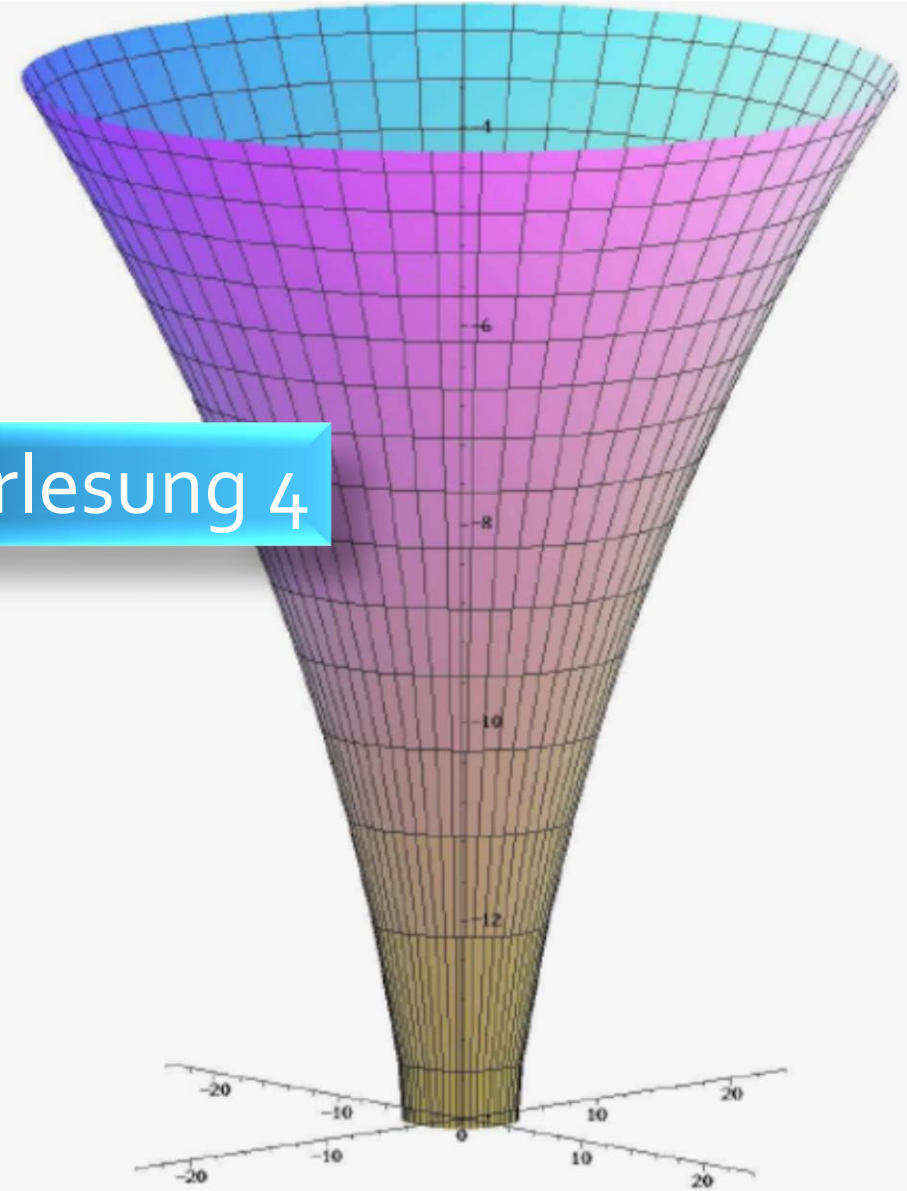
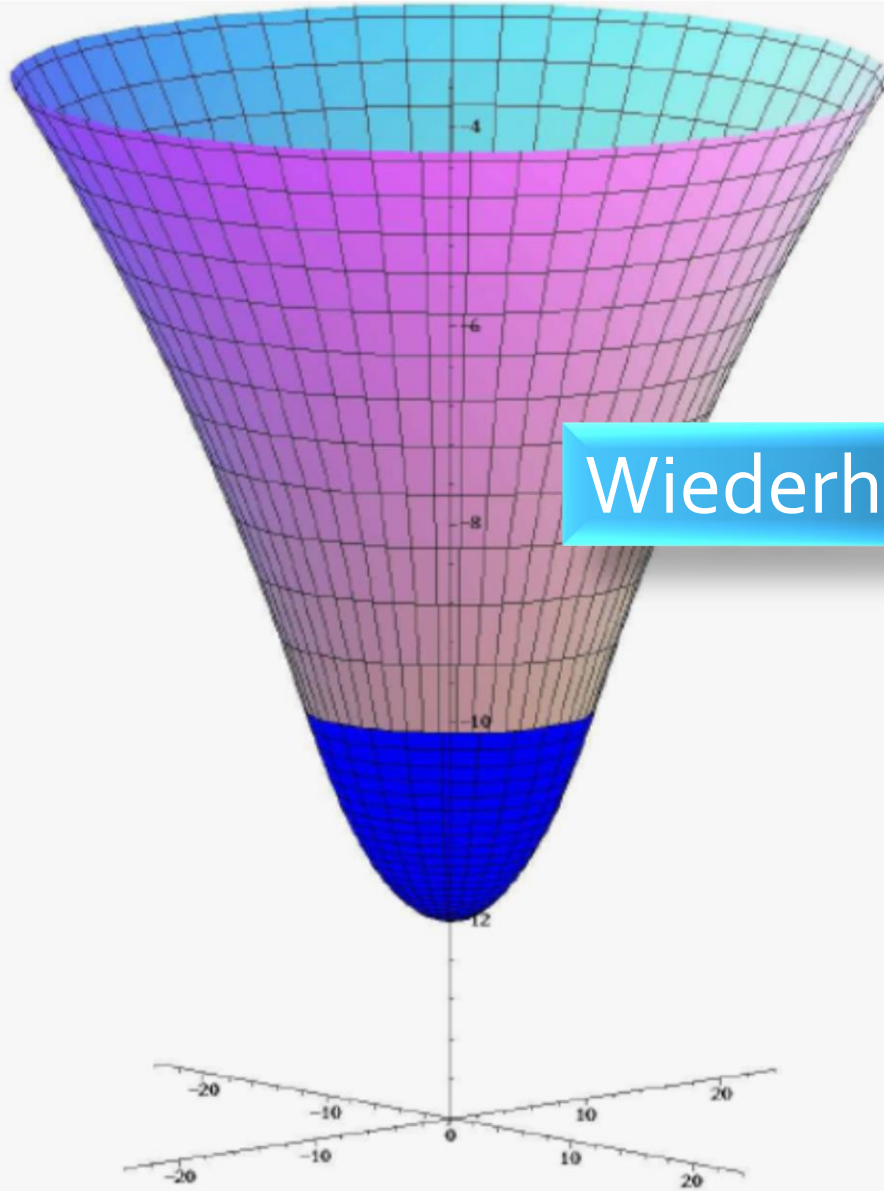
Aufgrund der Corona Krise findet die Vorlesung und die freiwilligen Übungstermine in diesem Semester nur Online statt.

## 5. Vorlesung

# Neutronenstern

# Schwarzes Loch

Wiederholung Vorlesung 4





# Die innere Schwarzschildlösung eines sphärisch symmetrischen, statischen Objektes (z.B. Erde, Neutronenstern)

Im folgenden wird die Einsteingleichung einer sphärisch symmetrischen und statischen Materieverteilung betrachtet. Die Materie wird hierbei als ideale Flüssigkeit angesetzt.

## Von der Einstein Gleichung zur Tolman-Oppenheimer-Volkoff Gleichung (TOV)

```
> restart:  
with( tensor );
```

Wir definieren einen sphärisch symmetrischen und statischen Ansatz der Metrik:

$$g_{\mu\nu} = \begin{pmatrix} e^{2\phi(r)} & 0 & 0 & 0 \\ 0 & -\left(1 - \frac{2m(r)}{r}\right)^{-1} & 0 & 0 \\ 0 & 0 & -r^2 & 0 \\ 0 & 0 & 0 & -r^2 \sin^2(\theta) \end{pmatrix} \quad \text{mit: } x^\mu = (t, r, \theta, \phi) \quad ,$$

wobei die Funktionen  $\phi(r)$  und  $m(r)$  an dieser Stelle noch unbekannt sind und keine physikalische Bedeutung besitzen.

```
> coord := [t, r, theta, phi]:  
g_compts := array(symmetric, sparse, 1..4, 1..4):  
g_compts[1,1] := exp(2*phi(r)):  
# a_compts[2,2] := exp(2*lambda(r)):
```

Der Energie-Impuls Tensor (rechte Seite der Einsteingleichung) wird als ideale Flüssigkeit angesetzt:

$$T^{\mu}_{\nu} = \begin{pmatrix} e(r) & 0 & 0 & 0 \\ 0 & -p(r) & 0 & 0 \\ 0 & 0 & -p(r) & 0 \\ 0 & 0 & 0 & -p(r) \end{pmatrix},$$

wobei die Funktionen  $e(r)$  und  $p(r)$  die Energiedichte und den Druck der Neutronensternmaterie darstellen, die ihrerseits über die Zustandsgleichung  $p(e)$  miteinander verknüpft sind.

```
> T:=create([1,-1], array([[e(r),0,0,0],[0,-p(r),0,0],[0,0,-p(r),0],[0,0,0,-p(r)]]));
    Tl:=lower(g,T,1);
    Tu:=raise(ginv,T,2);
    prod(ginv, Tl, [2, 1]);
    contract(T, [1, 2]);
```

$$T := \text{table} \left( \text{compts} = \begin{pmatrix} e(r) & 0 & 0 & 0 \\ 0 & -p(r) & 0 & 0 \\ 0 & 0 & -p(r) & 0 \\ 0 & 0 & 0 & -p(r) \end{pmatrix}, \text{index\_char} = [1, -1] \right)$$

$$Tl := \text{table} \left( \text{compts} = \begin{pmatrix} e^{2\Phi(r)} e(r) & 0 & 0 & 0 \\ 0 & -\frac{r p(r)}{-r + 2 m(r)} & 0 & 0 \\ 0 & 0 & r^2 p(r) & 0 \\ 0 & 0 & 0 & r^2 \sin^2(\theta) p(r) \end{pmatrix}, \text{index\_char} = [-1, -1] \right)$$

(2.1.4)



Wir definieren die Einsteingleichung in der folgenden Form:

$$G_{\mu\nu} + 8\pi T_{\mu\nu} = 0$$

> **Einsteingl:=lin\_com(G,8\*Pi,Tl);**

$$\begin{aligned}
 \text{Einsteingl} := \text{table} \left( \text{compts} = \right. & \left. \left[ \left[ -\frac{2 e^{2\phi(r)} \left( \frac{d}{dr} m(r) - 4 \pi e(r) r^2 \right)}{r^2}, 0, 0, 0 \right], \right. \right. \\
 & \left. \left[ 0, -\frac{2 \left( 2 r \left( \frac{d}{dr} \phi(r) \right) m(r) + m(r) - \left( \frac{d}{dr} \phi(r) \right) r^2 + 4 \pi r^3 p(r) \right)}{r^2 (-r + 2 m(r))}, 0, 0 \right], \right. \\
 & \left[ 0, 0, \frac{1}{r} \left( -\left( \frac{d}{dr} \phi(r) \right) r^2 + r \left( \frac{d}{dr} \phi(r) \right) m(r) + \left( \frac{d}{dr} m(r) \right) r - m(r) \right. \right. \\
 & \left. \left. - \left( \frac{d^2}{dr^2} \phi(r) \right) r^3 + \left( \frac{d}{dr} \phi(r) \right) \left( \frac{d}{dr} m(r) \right) r^2 + 2 \left( \frac{d}{dr} \phi(r) \right)^2 r^2 m(r) \right. \right. \\
 & \left. \left. + 2 \left( \frac{d^2}{dr^2} \phi(r) \right) r^2 m(r) - \left( \frac{d}{dr} \phi(r) \right)^2 r^3 + 8 \pi r^3 p(r) \right], 0 \right], \\
 & \left[ 0, 0, 0, -\frac{1}{r} \left( \left( \frac{d}{dr} \phi(r) \right) r^2 - \left( \frac{d}{dr} \phi(r) \right) r^2 \cos(\theta)^2 - r \left( \frac{d}{dr} \phi(r) \right) m(r) \right. \right. \\
 & \left. \left. + r \left( \frac{d}{dr} \phi(r) \right) m(r) \cos(\theta)^2 - \left( \frac{d}{dr} m(r) \right) r + \left( \frac{d}{dr} m(r) \right) r \cos(\theta)^2 + m(r) \right. \right. \\
 & \left. \left. - m(r) \cos(\theta)^2 + \left( \frac{d^2}{dr^2} \phi(r) \right) r^3 - \left( \frac{d^2}{dr^2} \phi(r) \right) r^3 \cos(\theta)^2 \right. \right. \\
 & \left. \left. - \left( \frac{d}{dr} \phi(r) \right) \left( \frac{d}{dr} m(r) \right) r^2 + \left( \frac{d}{dr} \phi(r) \right) \left( \frac{d}{dr} m(r) \right) r^2 \cos(\theta)^2 \right. \right. \\
 & \left. \left. - 2 \left( \frac{d}{dr} \phi(r) \right)^2 r^2 m(r) + 2 \left( \frac{d}{dr} \phi(r) \right)^2 r^2 m(r) \cos(\theta)^2 \right] \right. \\
 & \left. \right) \quad (2.1.5)
 \end{aligned}$$

Aus der Einsteingleichung folgt die Erhaltung des Energie-Impulses. Diese sogenannten hydrodynamischen Gleichungen (kovariante Erhaltung des Energie-Impulses) sind durch die folgenden vier Gleichungen definiert (Bemerke: in der Literatur wird die kovariante Ableitung mit unterschiedlichen Symbolen bezeichnet):

$$\nabla_{\mu} G^{\mu}_{\nu} = D_{\mu} G^{\mu}_{\nu} = G^{\mu}_{\nu}{}_{||\mu} = 0 \quad \rightarrow \quad \nabla_{\mu} T^{\mu}_{\nu} = 0 \quad .$$

wobei die kovariante Ableitung eines Tensors zweiter Stufe wie folgt definiert ist:

$$\nabla_{\alpha} T^{\mu}_{\nu} = \partial_{\alpha} T^{\mu}_{\nu} + \Gamma^{\mu}_{\alpha\rho} T^{\rho}_{\nu} - \Gamma^{\rho}_{\alpha\nu} T^{\mu}_{\rho} \quad ,$$

```
> DT:=cov_diff(T, coord, Cf2);  
DTa:=get_compts(contract(DT, [1, 3]))[2]=0;
```

$$DTa := -\left(\frac{d}{dr} \phi(r)\right) p(r) - \left(\frac{d}{dr} \phi(r)\right) e(r) - \left(\frac{d}{dr} p(r)\right) = 0 \quad (2.1.7)$$

und nach  $\frac{dp}{dr}$  aufgelöst ergibt sich das folgende:

$$\begin{aligned} \text{TOV1} &:= \frac{d}{dr} p(r) = \frac{(m(r) + 4 \pi r^3 p(r)) (e(r) + p(r))}{r (-r + 2 m(r))} \\ \text{TOV2} &:= \frac{d}{dr} m(r) = 4 \pi e(r) r^2 \\ \text{TOV3} &:= \frac{d}{dr} \phi(r) = - \frac{m(r) + 4 \pi r^3 p(r)}{r (-r + 2 m(r))} \end{aligned} \quad (2.1.10)$$

Das oben abgebildete System aus drei gekoppelten Differentialgleichungen bezeichnet man als die Tolman-Oppenheimer-Volkoff Gleichungen (TOV-Gleichung). Bemerkung: In manchen Büchern werden auch lediglich die ersten beiden Gleichungen als TOV-Gleichungen bezeichnet.



# Numerische Lösung der TOV-Gleichungen

Im folgenden werden die TOV-Gleichungen numerisch gelöst, indem wir einerseits eine Zustandsgleichung der Materie (eine Funktion  $p(e)$ ) festlegen und von einem Startwert der zentralen Energiedichte im Inneren des sphärisch symmetrischen Objektes nach Außen integrieren.

>

```
a:=10;  
b:=5/3;  
p(r):=a*(e(r))^b;  
W3:=plot(a*x^b,x=0..1,color=blue):  
TOV1:=solve(DTa,diff(phi(r), r))*(e(r)+p(r))*(-1)=rhs(Einstein2)*  
(e(r)+p(r))*(-1);  
TOV2:=Einstein1;  
TOV3:=Einstein2;
```

$$\begin{aligned} a &:= 10 \\ p(r) &:= 10 e(r)^{5/3} \\ TOV1 &:= \frac{50}{3} e(r)^{2/3} \left( \frac{d}{dr} e(r) \right) = \frac{(m(r) + 40 \pi r^3 e(r)^{5/3}) (10 e(r)^{5/3} + e(r))}{r (-r + 2 m(r))} \\ TOV2 &:= \frac{d}{dr} m(r) = 4 \pi e(r) r^2 \\ TOV3 &:= \frac{d}{dr} \phi(r) = -\frac{m(r) + 40 \pi r^3 e(r)^{5/3}}{r (-r + 2 m(r))} \end{aligned} \tag{2.2.1}$$

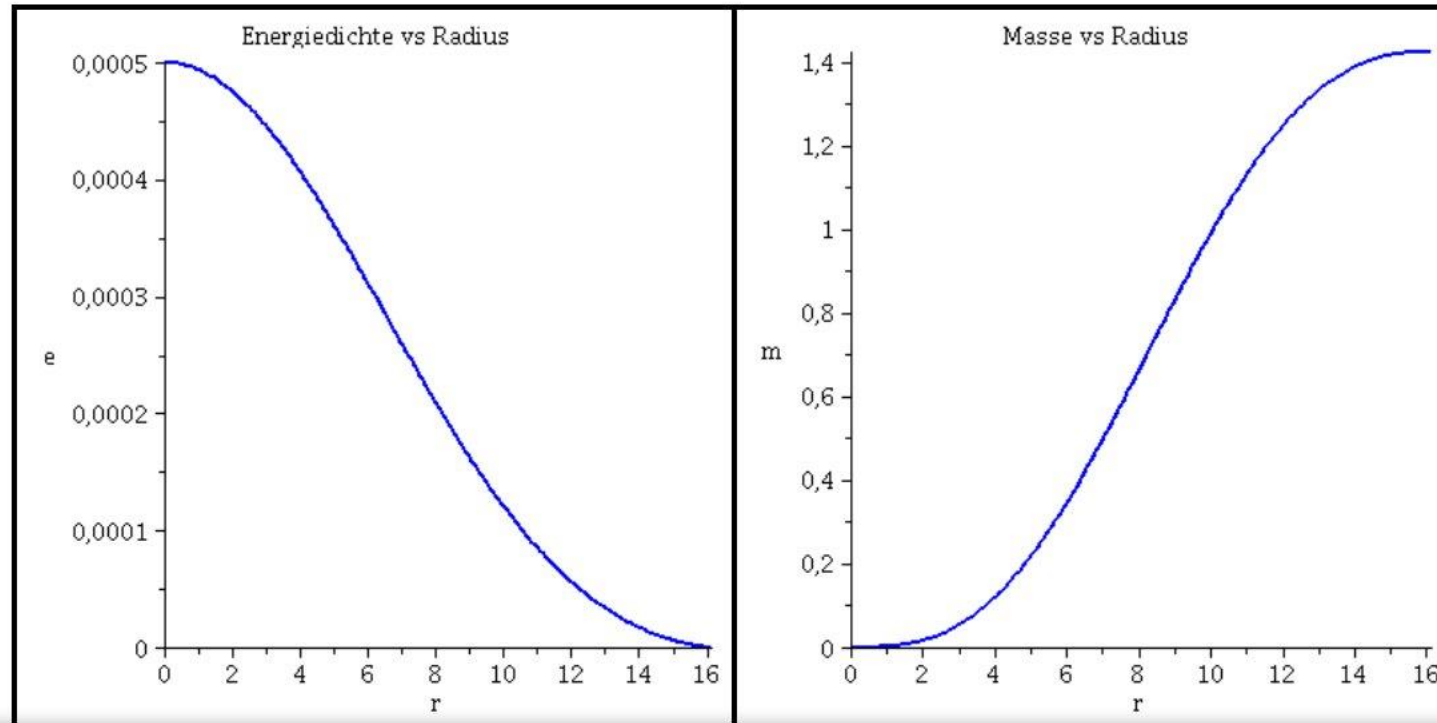
Numerische Lösung der Gleichung mit fixierten Randbedingungen im Sternzentrum.

>

```
r0:=10^(-14);  
e0:=0.0005;  
Loes:=dsolve({TOV1,TOV2,m(r0)=0,e(r0)=e0},  
{m(r),e(r)},type=numeric,output=listprocedure);
```

Darstellung der Energiedichte ( $e(r)$ ) und der Sternmasse ( $m(r)$ ) als Funktion des radialen Abstands  $r$  vom Sternzentrum:

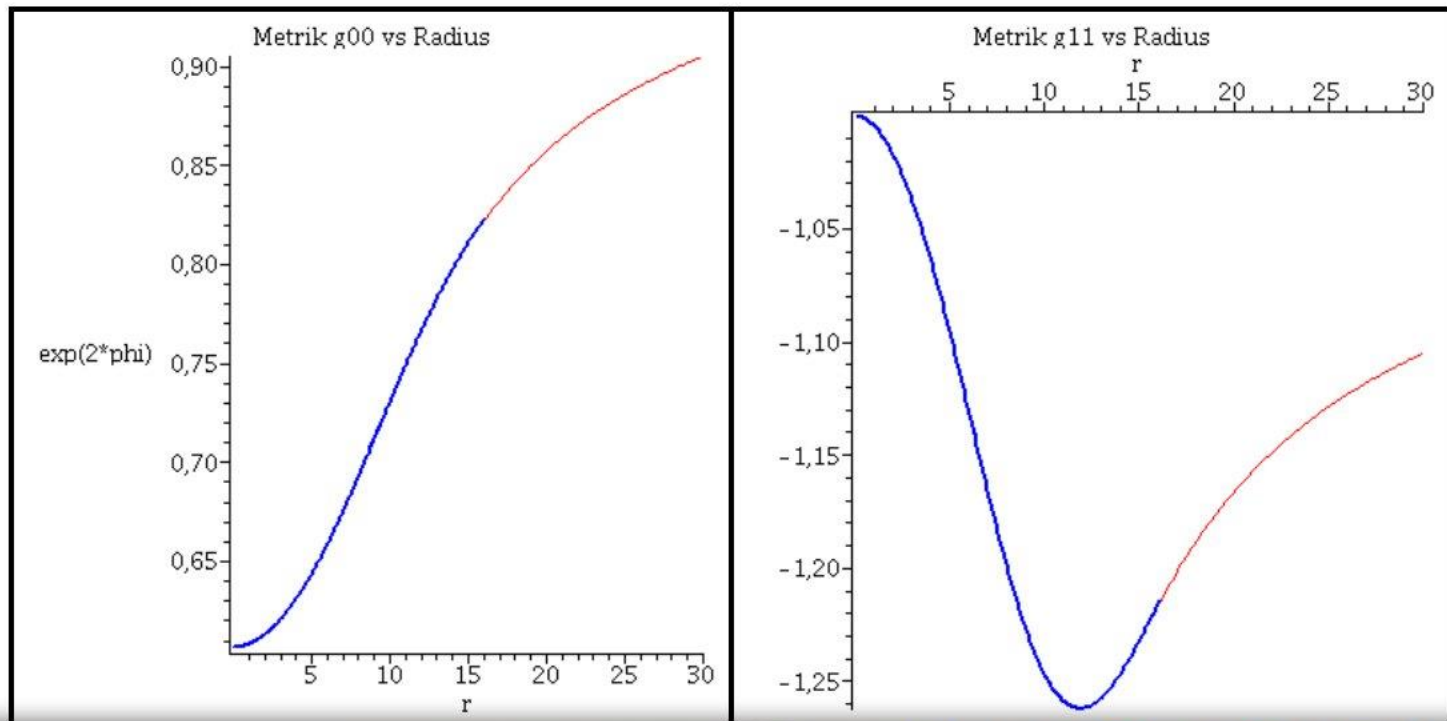
```
> with(plots):  
  rend:=16.12487:  
  Plot1:=odeplot(Loes,[r,e(r)],r0..rend,numpoints=200,color=blue,thickness=2,title="Energiedichte vs Radius"):  
  Plot2:=odeplot(Loes,[r,m(r)],r0..rend,numpoints=200,color=blue,thickness=2,title="Masse vs Radius"):  
  display(Matrix(1,2,[Plot1,Plot2]));
```



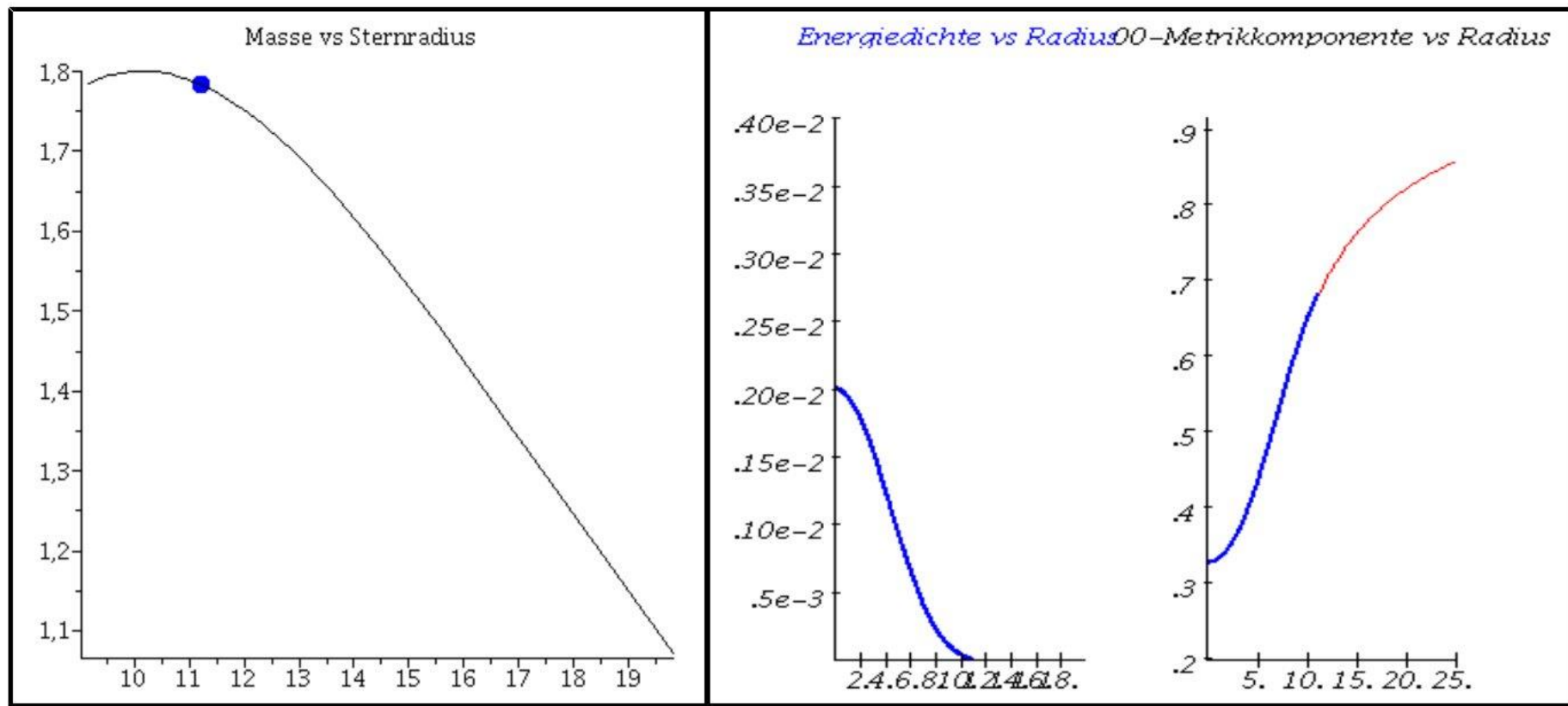


Veranschaulichung der  $g_{00}$ -Komponente (linke Abbildung) und  $g_{11}$ -Komponente (rechte Abbildung) der Innenraummetrik (blaue Kurven) und Außenraummetrik (rote Kurven).

```
> ranf:=10^(-1):  
Plot3:=odeplot(Loes1,[r,exp(2*phi(r))],ranf..rend,numpoints=200,color=blue,thickness=2,title="Metrik g00 vs Radius"):  
Plot4:=odeplot(Loes1,[r,-1/(1-2*m(r)/r)],ranf..rend,numpoints=200,color=blue,thickness=2,title="Metrik g11 vs Radius"):  
Plot5:=plot(1-2*M/r,r=rend..30,color=red):  
Plot6:=plot(-1/(1-2*M/r),r=rend..30,color=red):  
display(Matrix(1,2,[[display(Plot3,Plot5),display(Plot4,Plot6)]]));
```



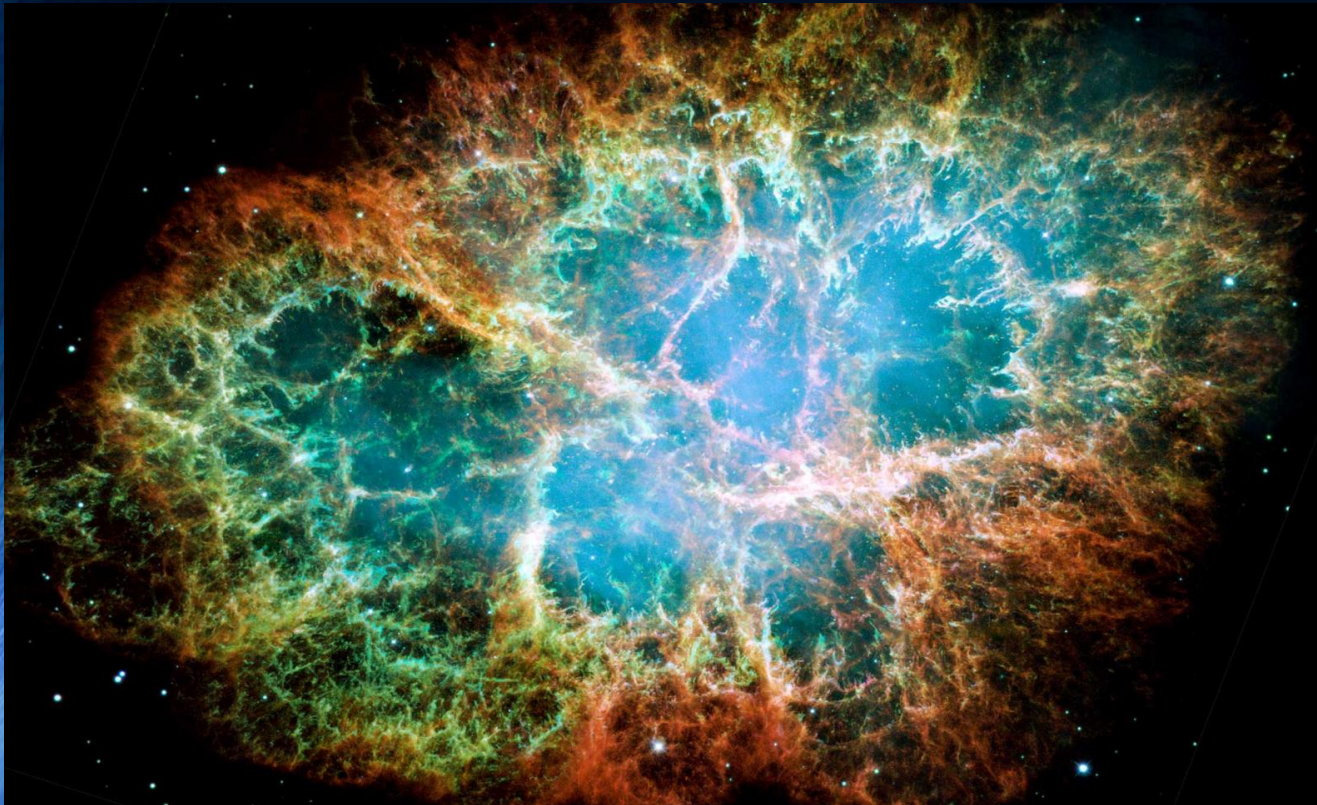
```
> Animat:=display([seq(Ani1[i],i=1..frames)],insequence=true):  
  Animat1:=display([seq(Ani2[i],i=1..frames)],insequence=true):  
  display(Array([Animat1,Animat]));
```





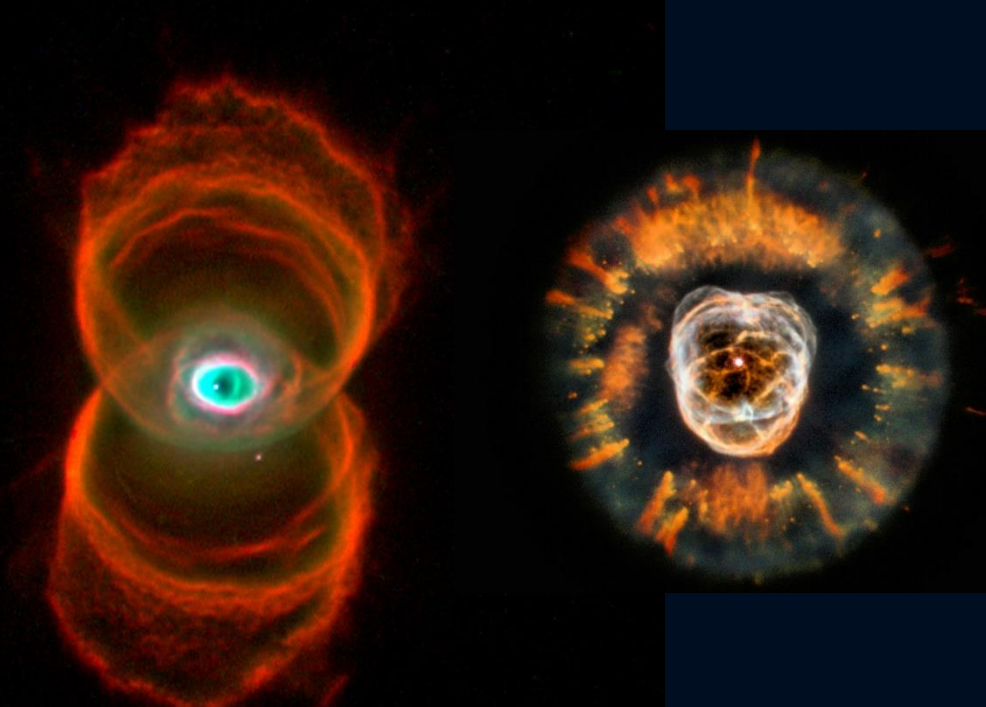
# Was sind Neutronensterne?

Neutronensterne entstehen in einer Supernova Explosion. Sonnen, die mindestens 8-mal schwerer als unsere Sonne sind explodieren am Ende ihrer Lebenszeit in einer Supernova Explosion – im Zentrum bleibt ein Neutronestern oder ein schwarzes Loch zurück.



Supernova Explosion, Krabben-Nebel

Zwei planetarische Nebel  
Endstadium leichter Sonnen (weißer Zwerg)

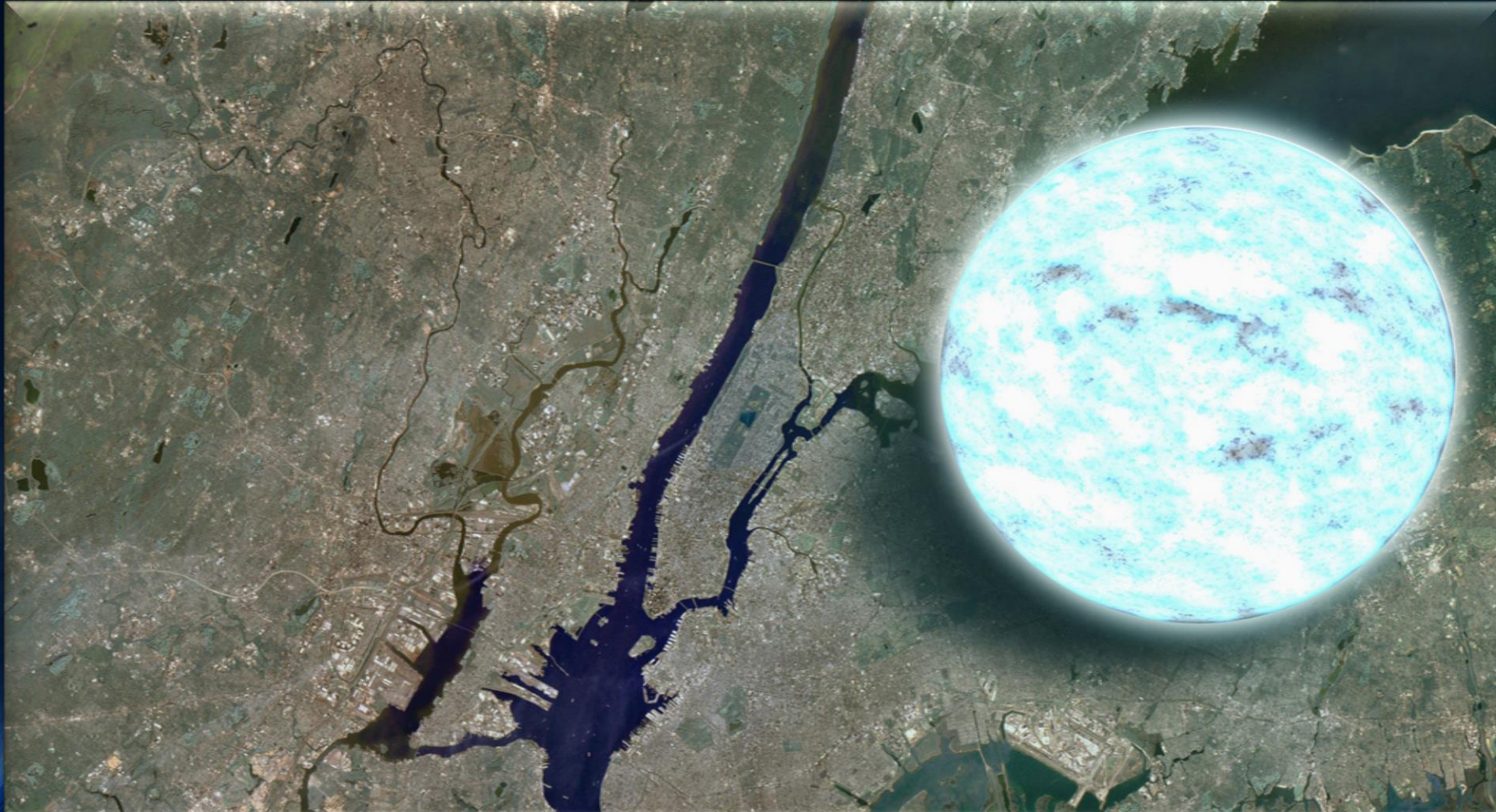




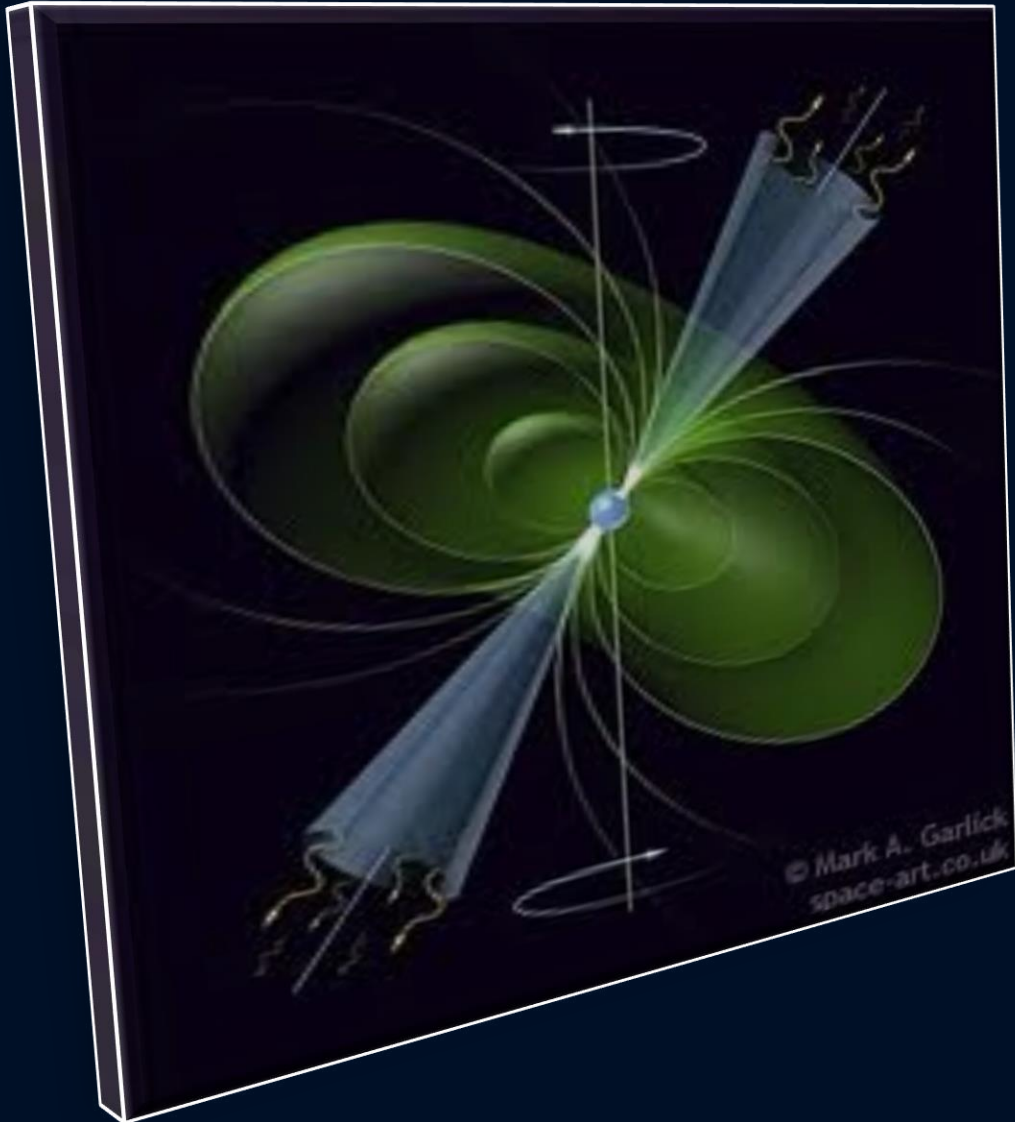
# Neutronensterne: Sehr klein und sehr schwer

Radius ~ 10 km, Masse ~ 1-2 Sonnenmassen

Riesige Magnetfelder ~  $10^{11}$  Tesla, schnell rotierend (bis zu 716 Hz)



# Pulsare := Rotierende Neutronensterne mit starkem Magnetfeld



In den letzten 50 Jahren konnten mittels Radioteleskopen ca. 3000 rotierende Neutronensterne (Pulsare) gefunden werden.

Der erste Pulsar wurde im Jahre 1967 entdeckt (PSR 1919+21, Bell)

Man unterscheidet Sekundenpulsare und Millisekunden-Pulsare



**PSR B0329+54 (0.715 s)**



**PSR B0531+21 (33.5 ms)**

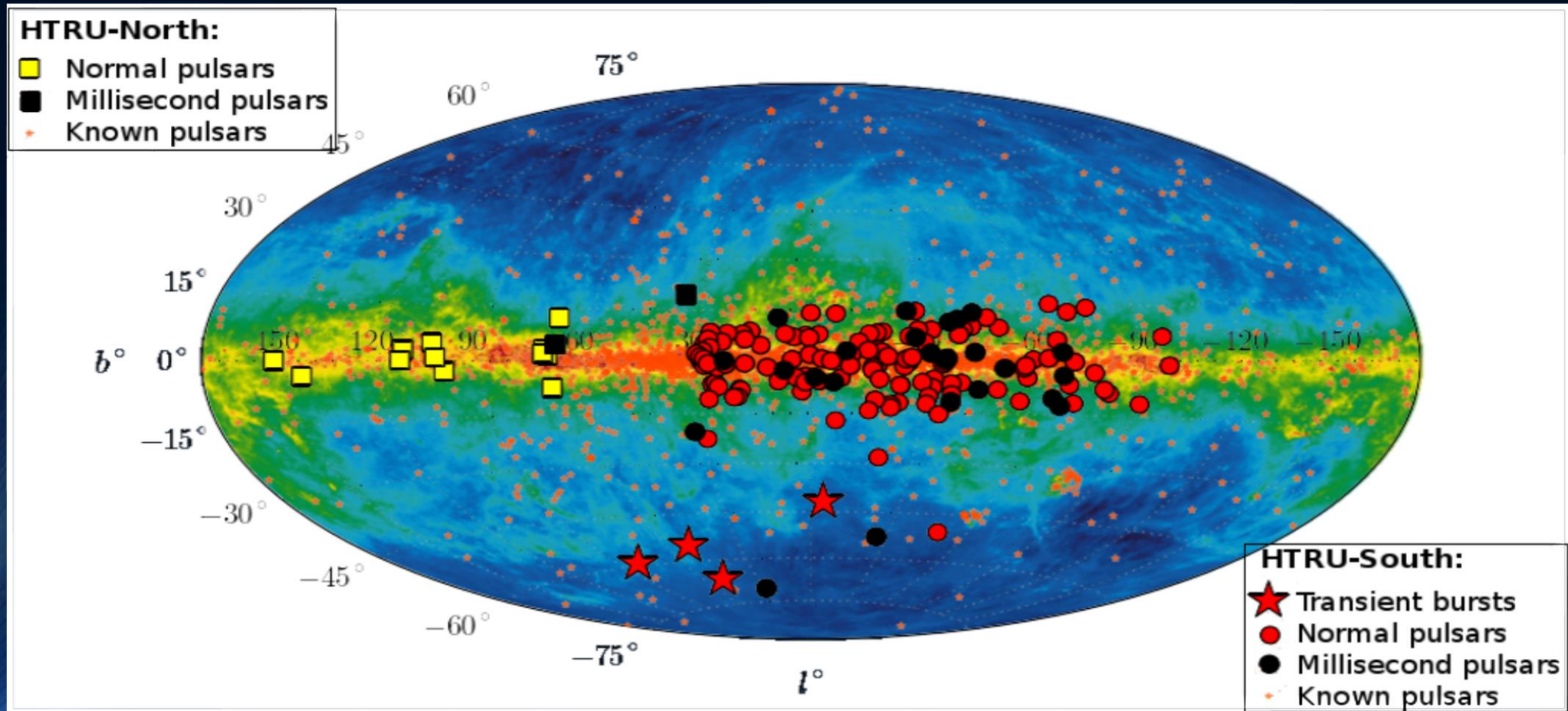


**PSR B1937+21 (1.56 ms)**



# Pulsare sind rotierende Neutronensterne

Zurzeit kennen wir ca. 3000 Neutronensterne





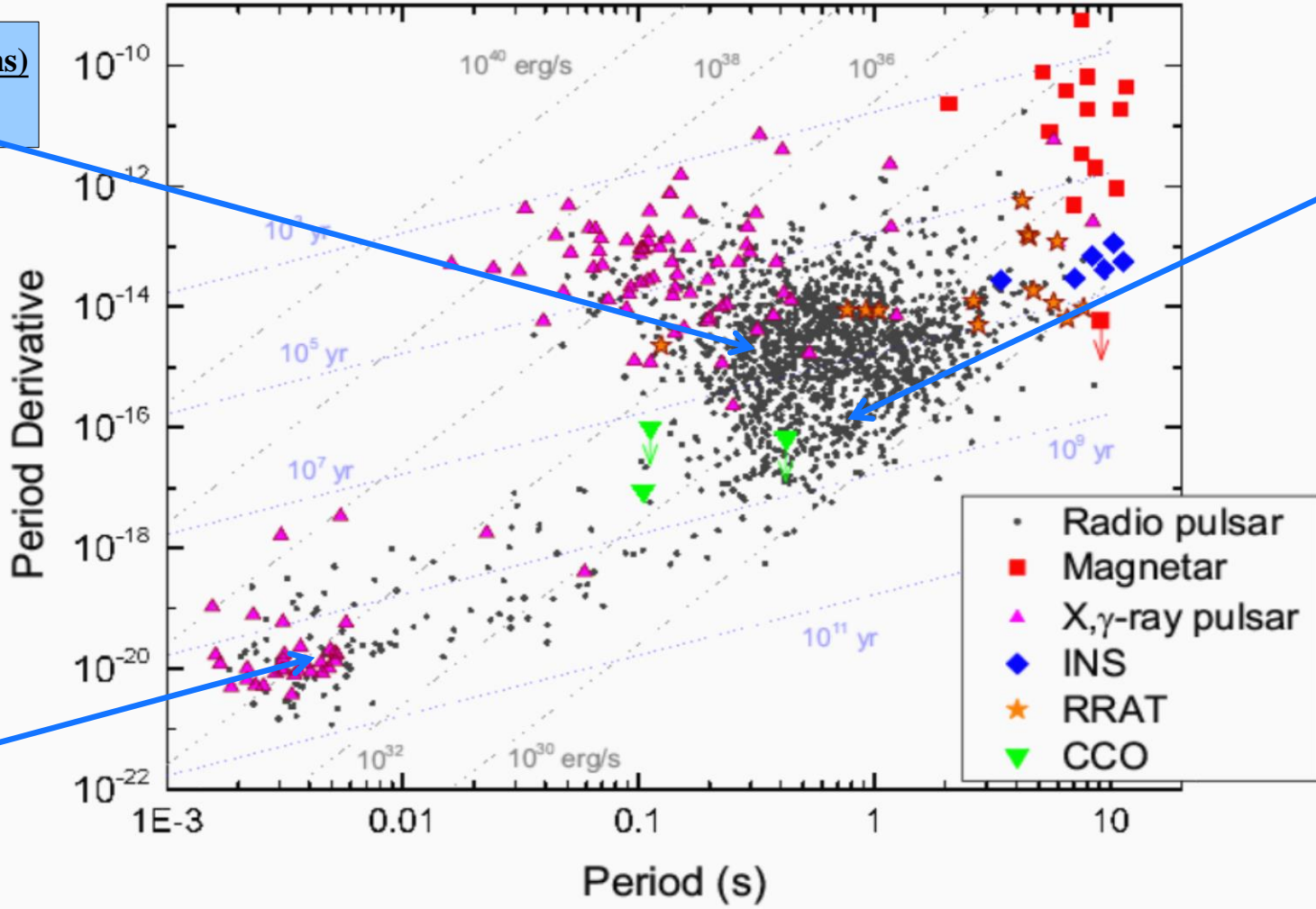
# Millisekunden und Sekunden Pulsare



**PSR B0531+21 (33.5 ms)**  
**Crab Pulsar**



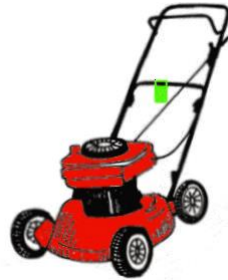
**PSR B0329+54 (0.715 s)**



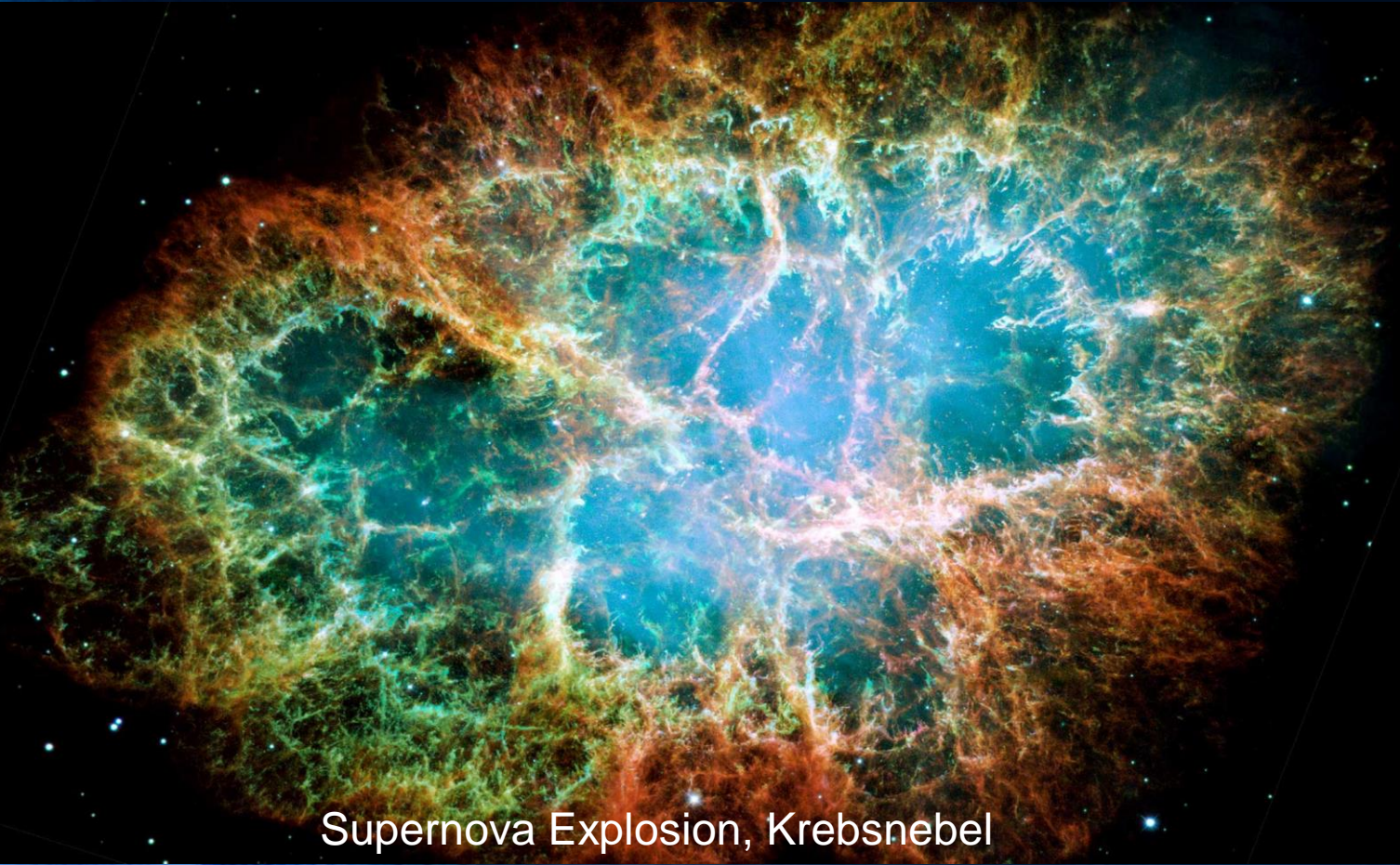
**PSR B1937+21 (1.56 ms)**

# Wie entstehen Neutronensterne?

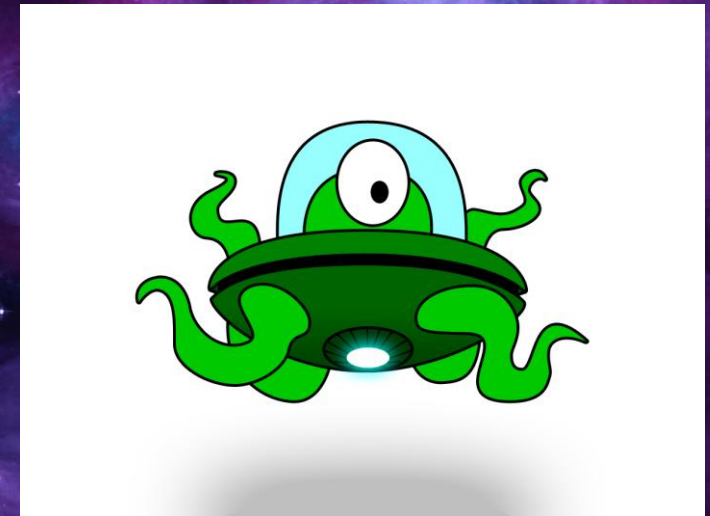
Neutronensterne entstehen in einer Supernova Explosion. Sonnen, die mindestens 8-mal schwerer als unsere Sonne sind explodieren am Ende ihrer Lebenszeit in einer Supernova Explosion – im Zentrum bleibt ein Neutronenstern oder ein schwarzes Loch zurück.



Im Zentrum des Nebels  
ist ein Neutronenstern



Supernova Explosion, Krebsnebel



Krebsnebel (Röntgenteleskop Chandra )



# Pulsare sind Rotierende Neutronensterne mit starkem Magnetfeld



Radioteleskop FAST in China



Radioteleskop in Effelsberg

Der erste Pulsar wurde im Jahre 1967 entdeckt (PSR 1919+21, Jocelyn Bell) und wurde zunächst LGM-1 genannt.



In den letzten 50 Jahren konnten mittels Radioteleskopen ca. 3000 rotierende Neutronensterne (Pulsare) gefunden werden.

Man unterscheidet  
Sekundenpulsare  
und  
Millisekunden-Pulsare



PSR B0329+54 (0.715 s)



PSR B0531+21 (33.5 ms)



PSR B1937+21 (1.56 ms)



# Beobachtete Massen von Neutronensternen in binären Systemen

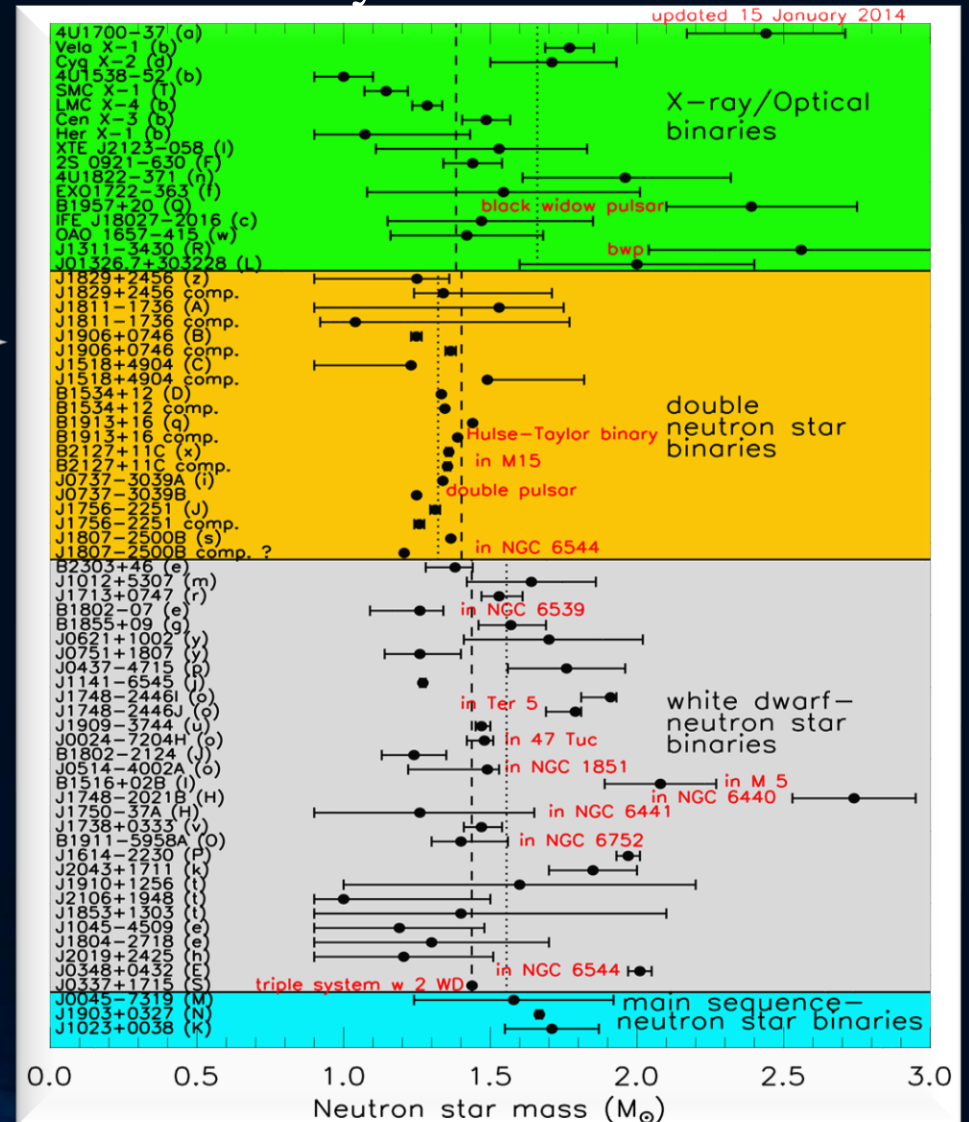
Einige der bekannten Neutronensterne befinden sich in Zweiersystemen:

NS-Planet, NS-(weißer Zwerg) oder NS-NS Systeme

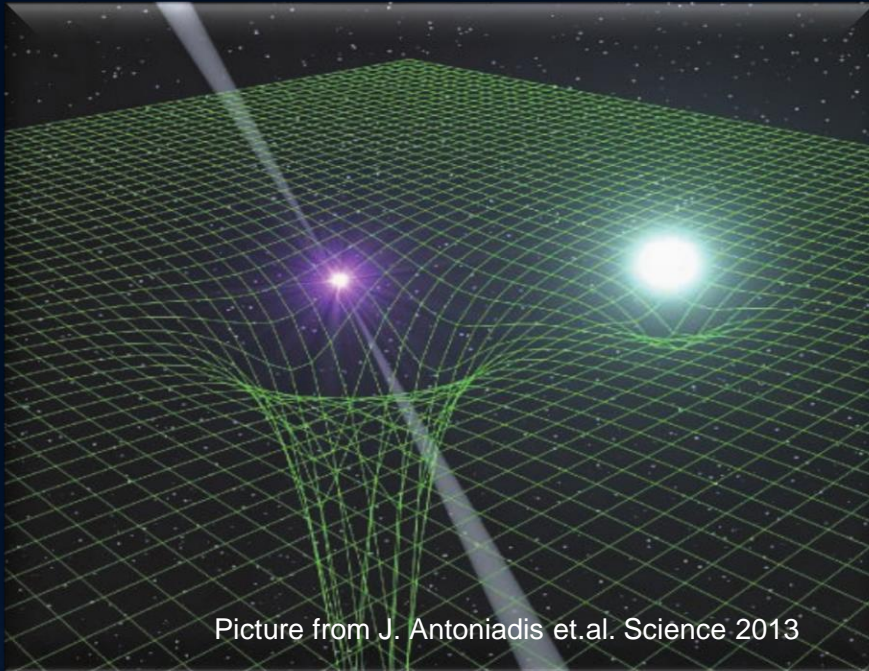
## PSR J1906+0746

144-ms Pulsar, entdeckt in 2004  
Orbitale Periode: 3.98 Stunden,  
Exzentrizität: 0.085  
Pulsar Masse: 1.291(11)  
Masse Begleiter: 1.322(1)  
In den Jahren 1998-2009 beobachtet,  
dann verschwand  
der Pulsar aufgrund der  
Spin-Präzession

Van Leeuwen et al, arXiv:1411.1518



# Beobachtete Massen von Neutronensternen in binären Systemen



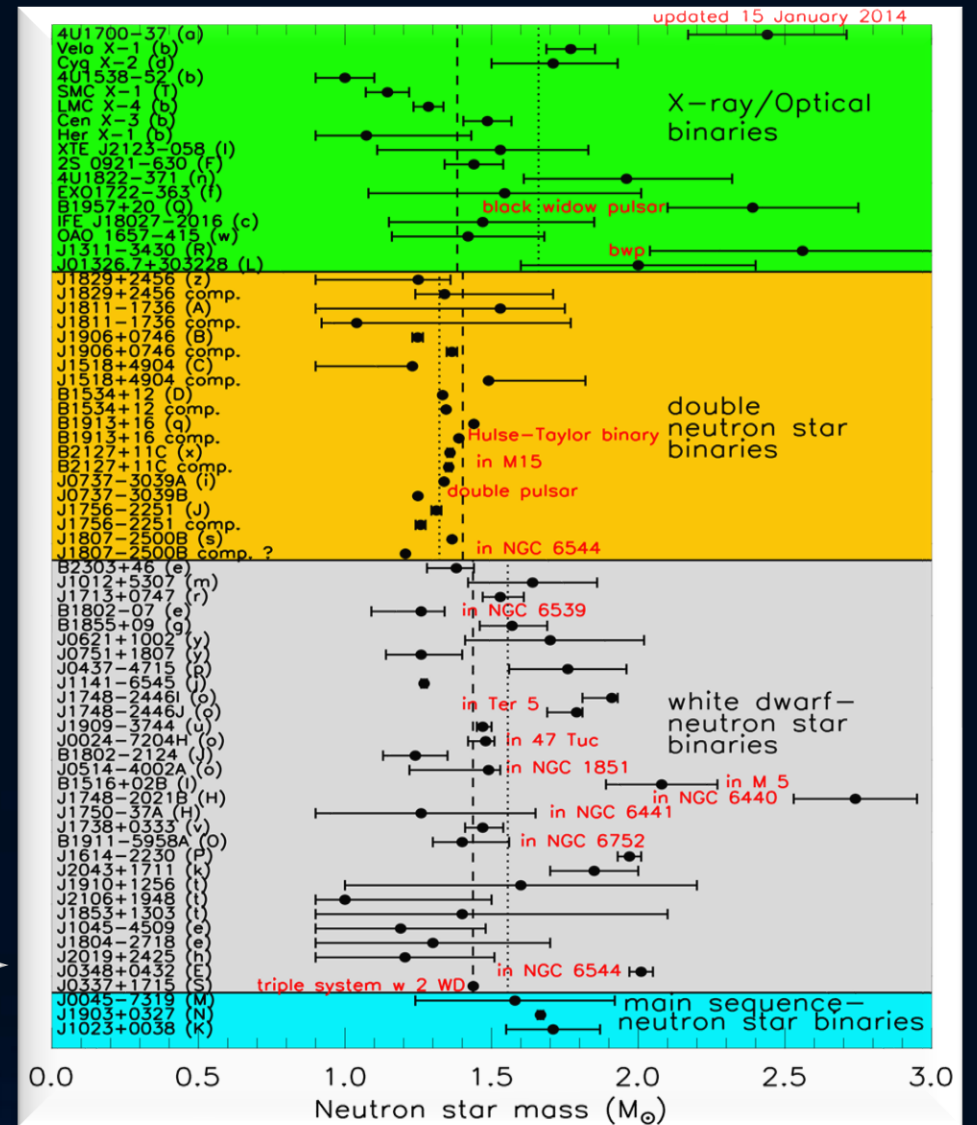
## PSR J0348+0432

Orbitale Periode: 2.46 Stunden

Pulsar mass:  $2.01 \pm 0.04$

Masse weißer Zwerg:

$M = 0.172 \pm 0.003$



# Binary Neutron Star Systems

Recently some new interesting Neutron Star Binary Systems has been found:

J0453+1559

$P = 17 \text{ ms}$

(similar to the Doublepulsar)

J1913+1102

$P = 27 \text{ ms}$

$P_b = 4.95 \text{ h}$

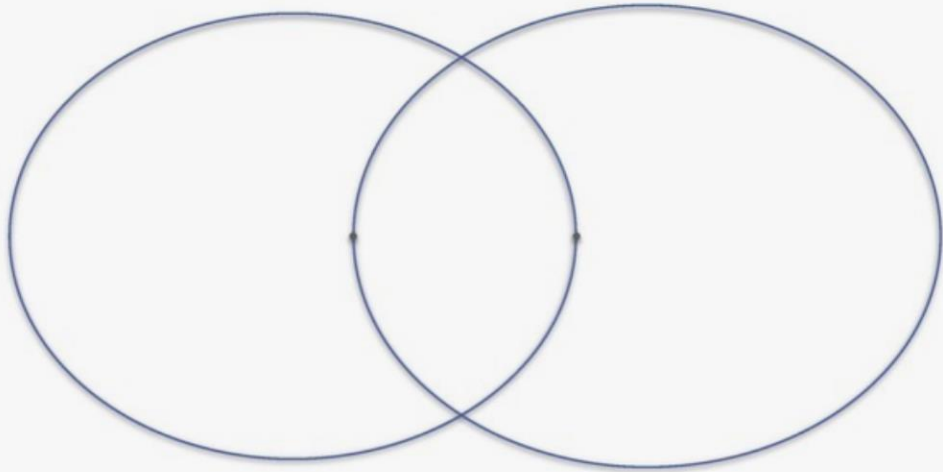
J1757-1854

$P = 215 \text{ ms}$

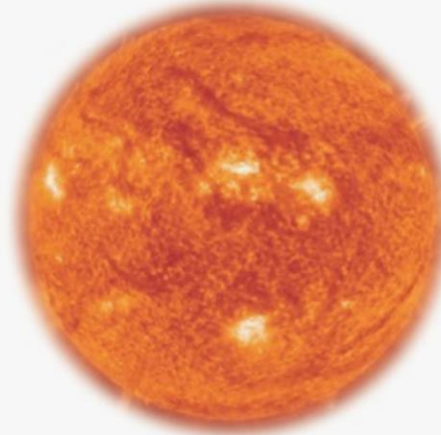
$P_b = 4.4 \text{ h}$

$E = 0.606$

Hulse-Taylor-Pulsar



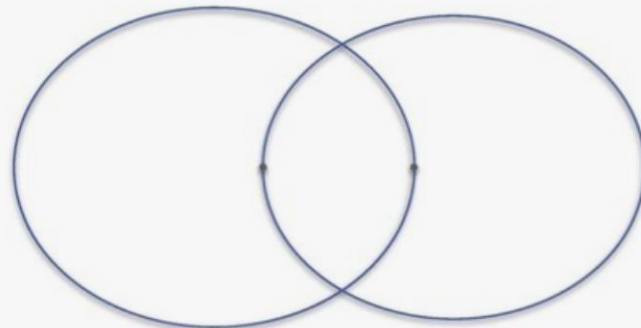
Sonne



Doppelpulsar



J1757-1854



Currently we know ~25 Double-NS Systems and one triple System



# Binäre Neutronenstern Systeme

Zurzeit kennt man ca. 25  
binäre Neutronenstern Systeme

Beispiel:

Der **Double Pulsar**

(PSR J0737-3039A/B):

Entdeckt im Jahre 2003

Eccentricity: 0.088

Pulsar A:  $P=23$  ms,  $M=1.3381(7)$

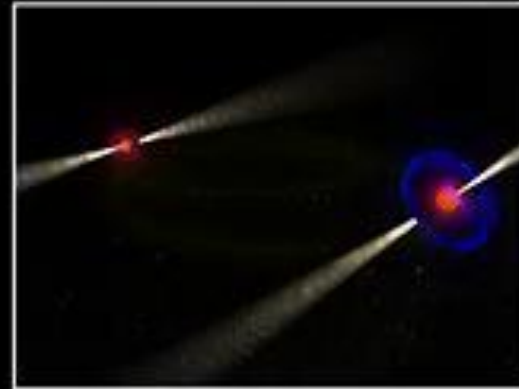
Pulsar B:  $P=2.7$  s,  $M=1.2489(7)$

Abstand zwischen den Sternen nur  
800,000 km

Orbitale Periode: 147 Minuten

Abstand verkleinert sich langsam  
aufgrund der Abstrahlung von  
Gravitationswellen

Die beiden Neutronensterne  
werden erst in 85 Millionen  
Jahren kollidieren



# The Einstein Equation

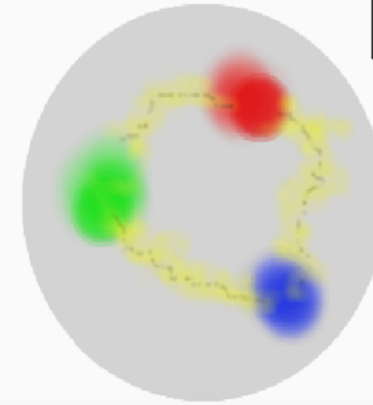
ART	<u>Yang-Mills-Theories</u>
$D_\beta v^\alpha = \partial_\beta v^\alpha + \Gamma_{\sigma\beta}^\alpha v^\sigma$	$D_{\beta a}{}^b = \partial_\beta 1_a{}^b + ig A_{\beta a}{}^b$
$R^\delta{}_{\mu\alpha\beta} v^\mu = [D_\alpha, D_\beta] v^\delta$	$F_{\alpha\beta a}{}^b = \frac{1}{ig} [D_{\alpha a}{}^c, D_{\beta c}{}^b]$
$R^\delta{}_{\mu\alpha\beta} = \Gamma_{\mu\alpha \beta}^\delta - \Gamma_{\mu\beta \alpha}^\delta$ $+ \Gamma_{\nu\beta}^\delta \Gamma_{\mu\alpha}^\nu + \Gamma_{\nu\alpha}^\delta \Gamma_{\mu\beta}^\nu$	$= A_{\beta a}{}^b _\alpha - A_{\alpha a}{}^b _\beta$ $+ \frac{1}{ig} [A_{\alpha a}{}^c, A_{\beta c}{}^b]$
$\mathcal{L}_G = R + \underbrace{(c_1 R_{\mu\nu} R^{\mu\nu} + \dots)}_{\equiv 0 \text{ for ART}}$	$\mathcal{L}_{YM} = \frac{1}{4} F_{\mu\nu a}{}^b F^{\mu\nu}{}_a{}^b$

Quantum ChromoDynamic:

( $SU(3)_{(c)}$ - Color Yang-Mills-Gauge Theory)

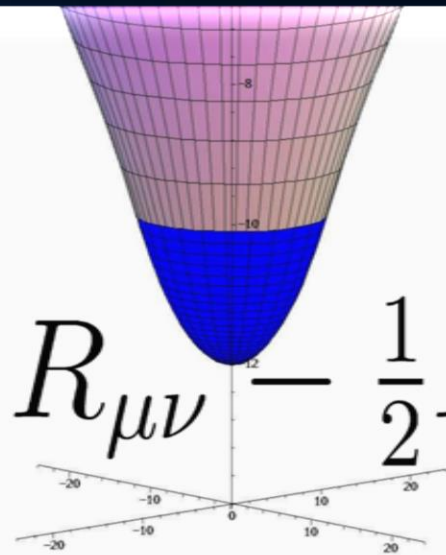
$$D_{\beta A}{}^B = \partial_\beta 1_A{}^B + ig G_{\beta A}{}^B$$

$A, B = \text{red, green, blue}$



$$\psi_A^f = \begin{pmatrix} \psi_r^f \\ \psi_g^f \\ \psi_b^f \end{pmatrix}$$

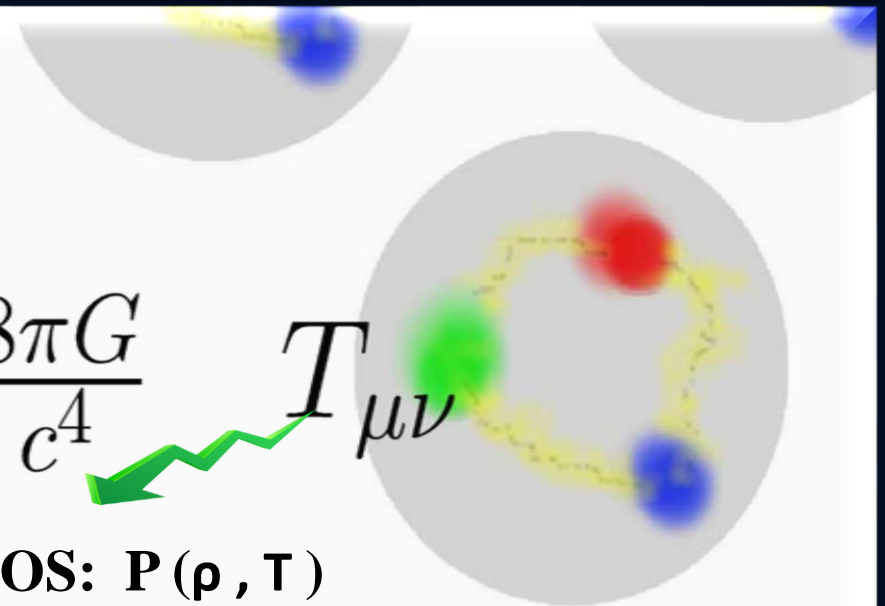
Confinement  
chiral symmetry, ...



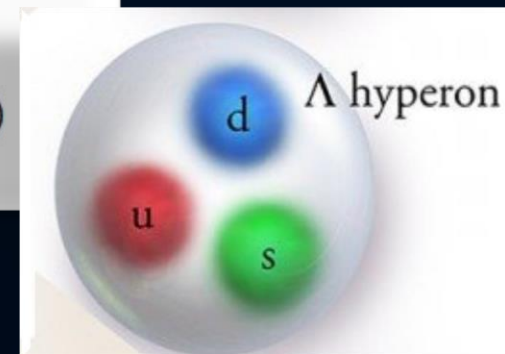
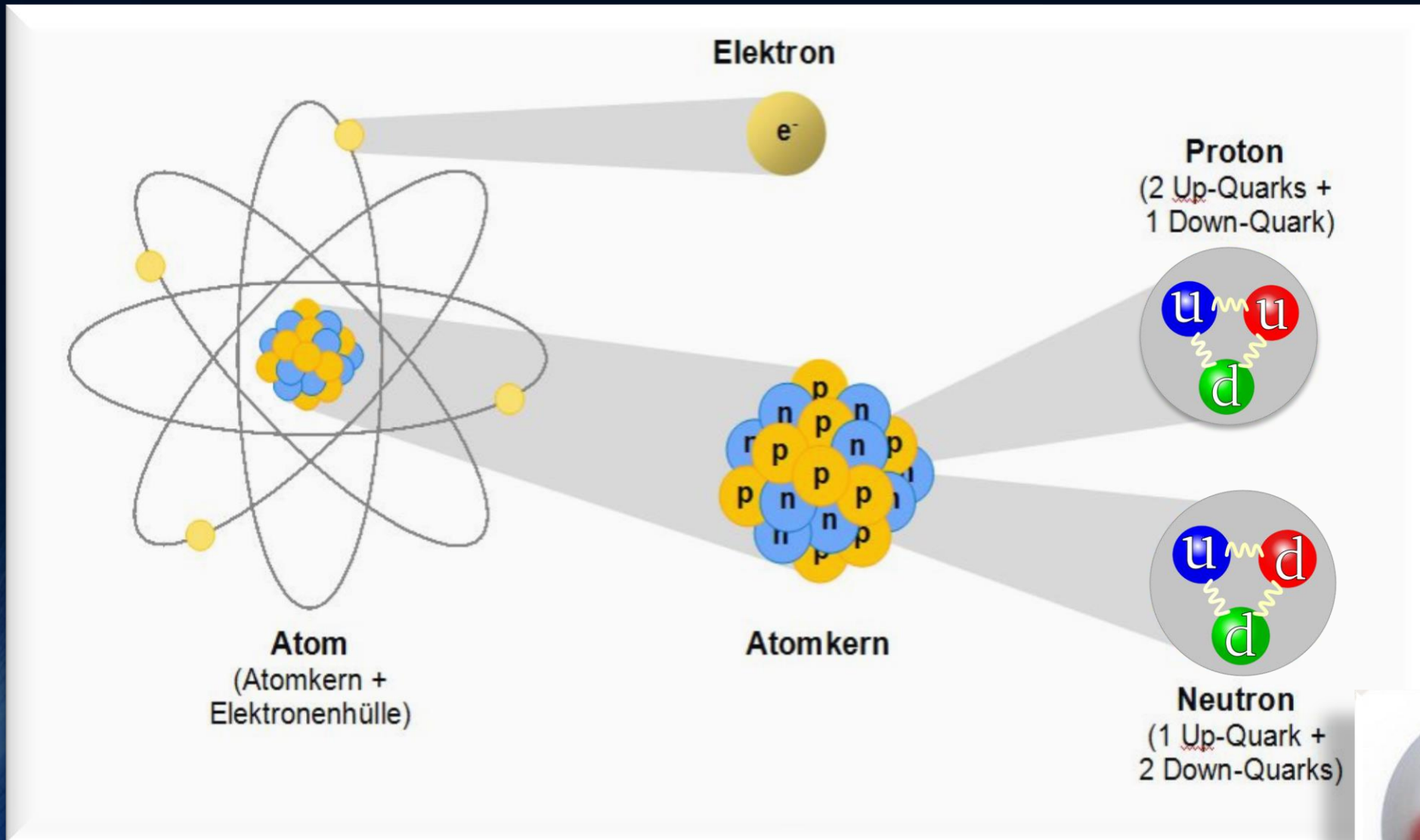
$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} =$$

$$\frac{8\pi G}{c^4} T_{\mu\nu}$$

EOS:  $P(\rho, T)$



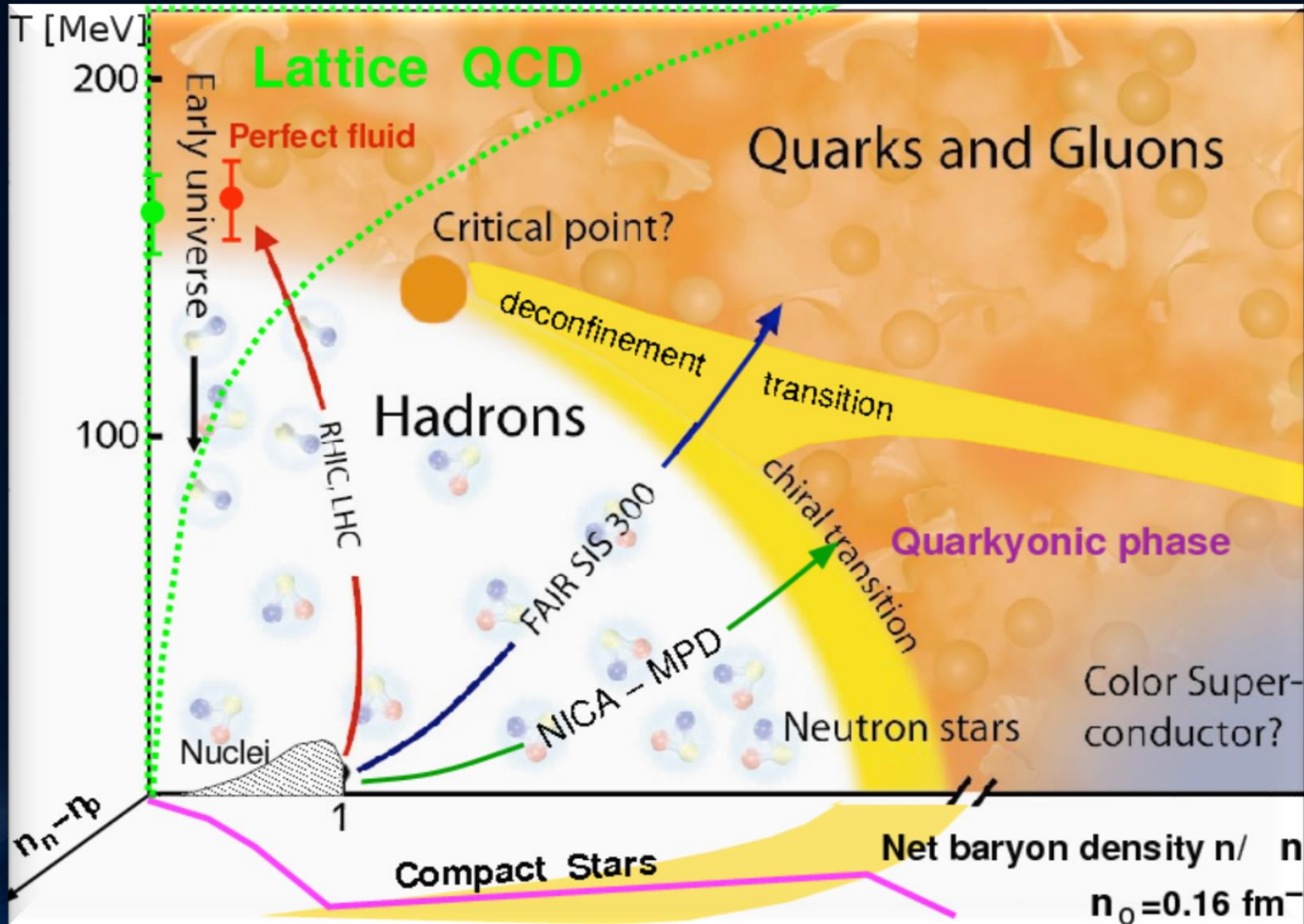
# Elementare Materie



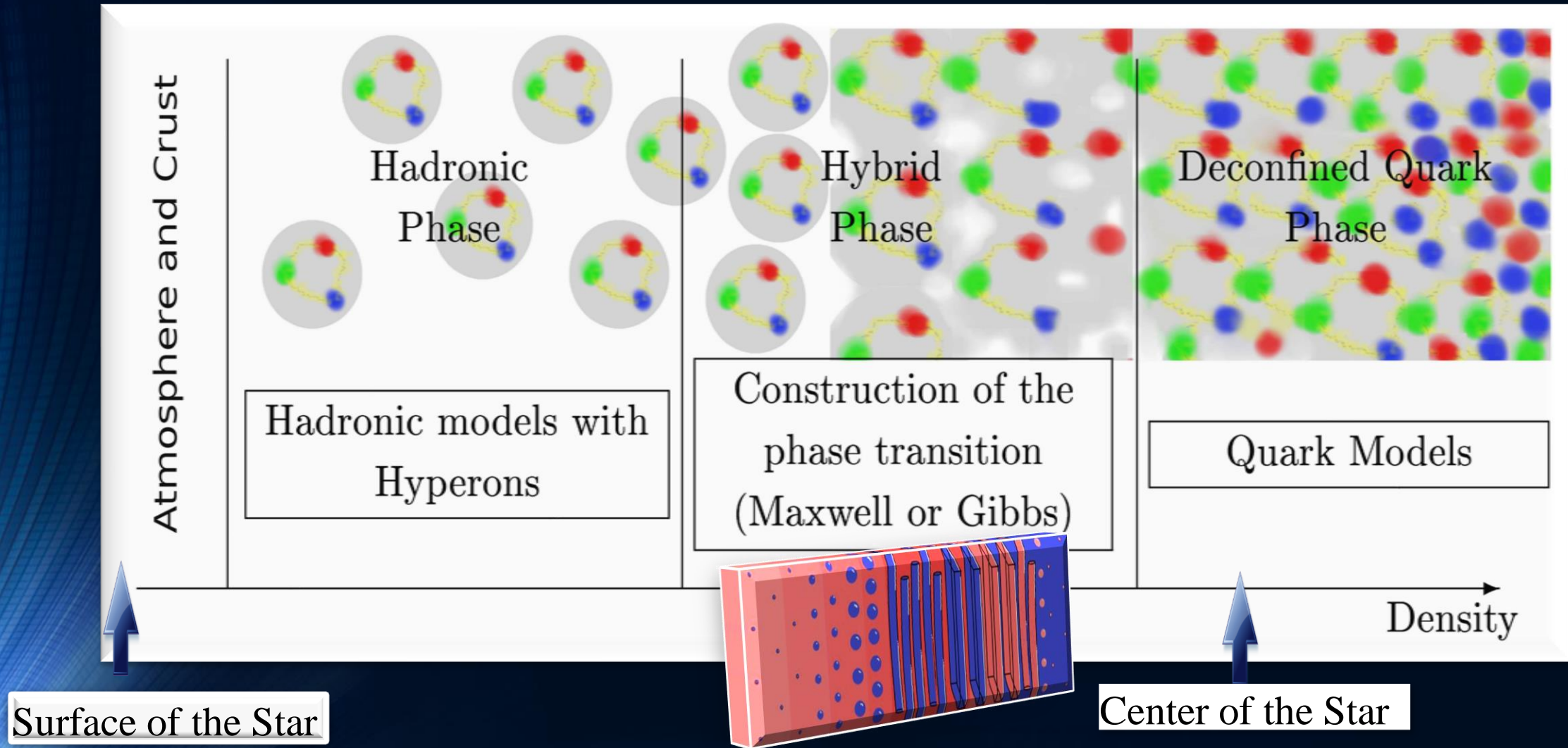
Bei sehr hohen Dichten oder Temperaturen können auch **Hyperonen** entstehen



# Die Zustandsgleichung der Materie und das Quark-Gluon-Plasma



# The QCD – Phase Transition and the Interior of a Hybrid Star

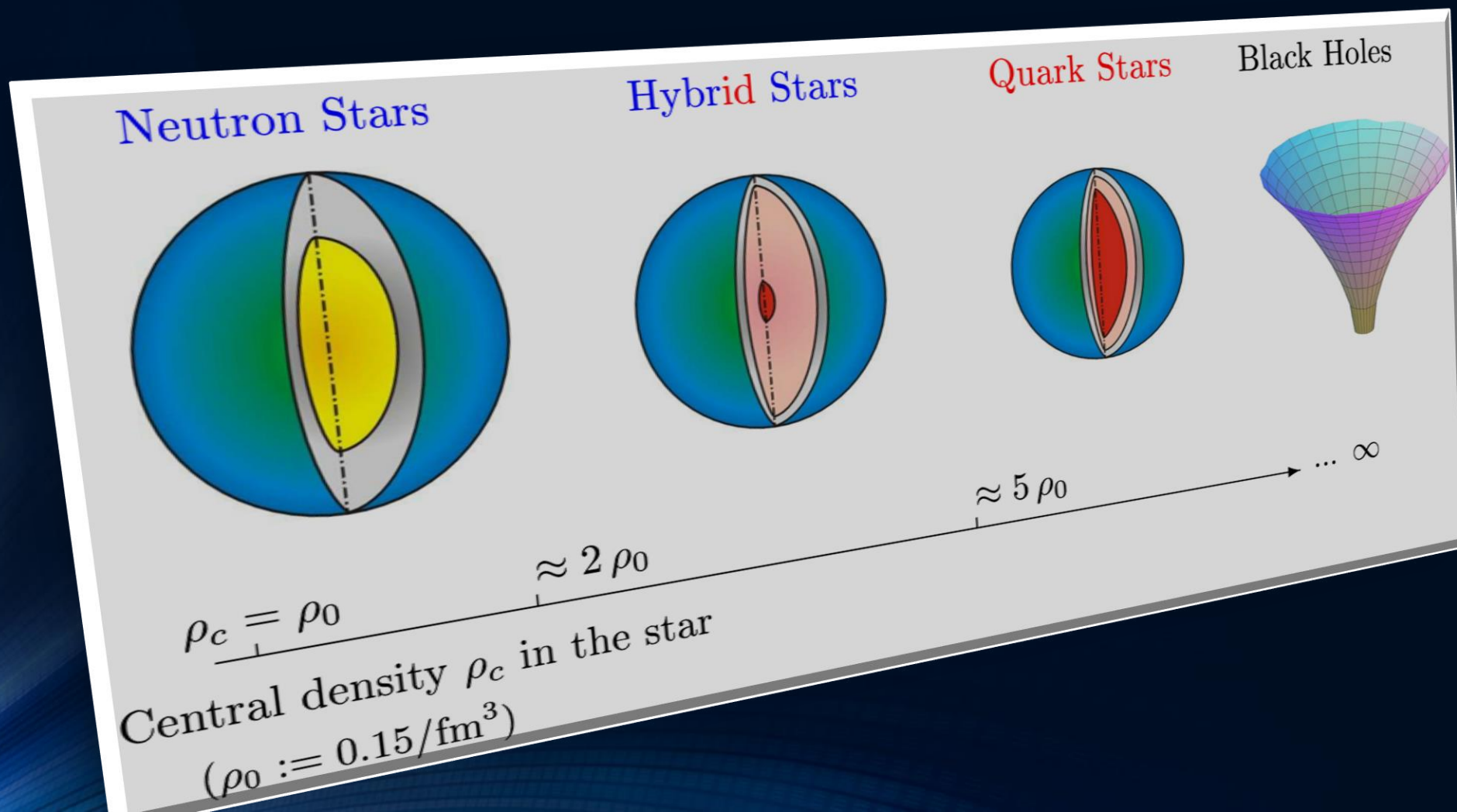


See: *Stable hybrid stars within a SU(3) Quark-Meson-Model*,  
A.Zacchi, M.Hanuske, J.Schaffner-Bielich, PRD 93, 065011 (2016)



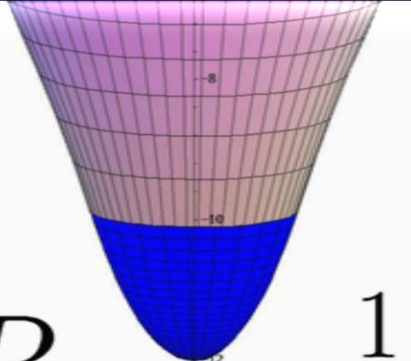
# Neutronensterne, Quarksterne und schwarze Löcher

Bei welcher Dichte der Phasenübergang zum Quark-Gluon-Plasma einsetzt und welche Eigenschaften dieser Übergang im Detail hat ist weitgehend unbekannt. Theoretische Modellierung mittels unterschiedlicher effektiver Elementarteilchenmodelle.

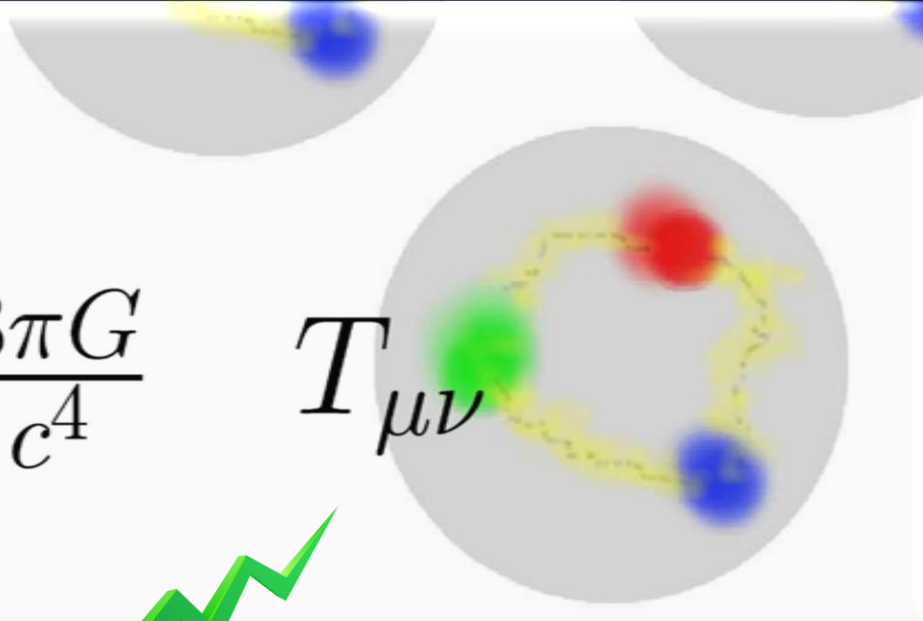


# Die Einstein Gleichung

Vor etwa 100 Jahren präsentierte Albert Einstein die Grundgleichung der Allgemeinen Relativitätstheorie  
– die sogenannte **Einstein-Gleichung**:



A diagram showing a grid of spacetime being curved downwards into a funnel shape, representing the effect of gravity. The grid is colored with a gradient from purple at the top to blue at the bottom.

$$R_{\mu\nu} - \frac{1}{2}R g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$


A diagram showing a circular cross-section of a system with a yellow dashed line representing a path or boundary. Inside the circle, there are several colored spheres (red, green, blue) representing matter or energy distribution.

Raumzeitkrümmung

Eigenschaften der Metrik  
der Raumzeit

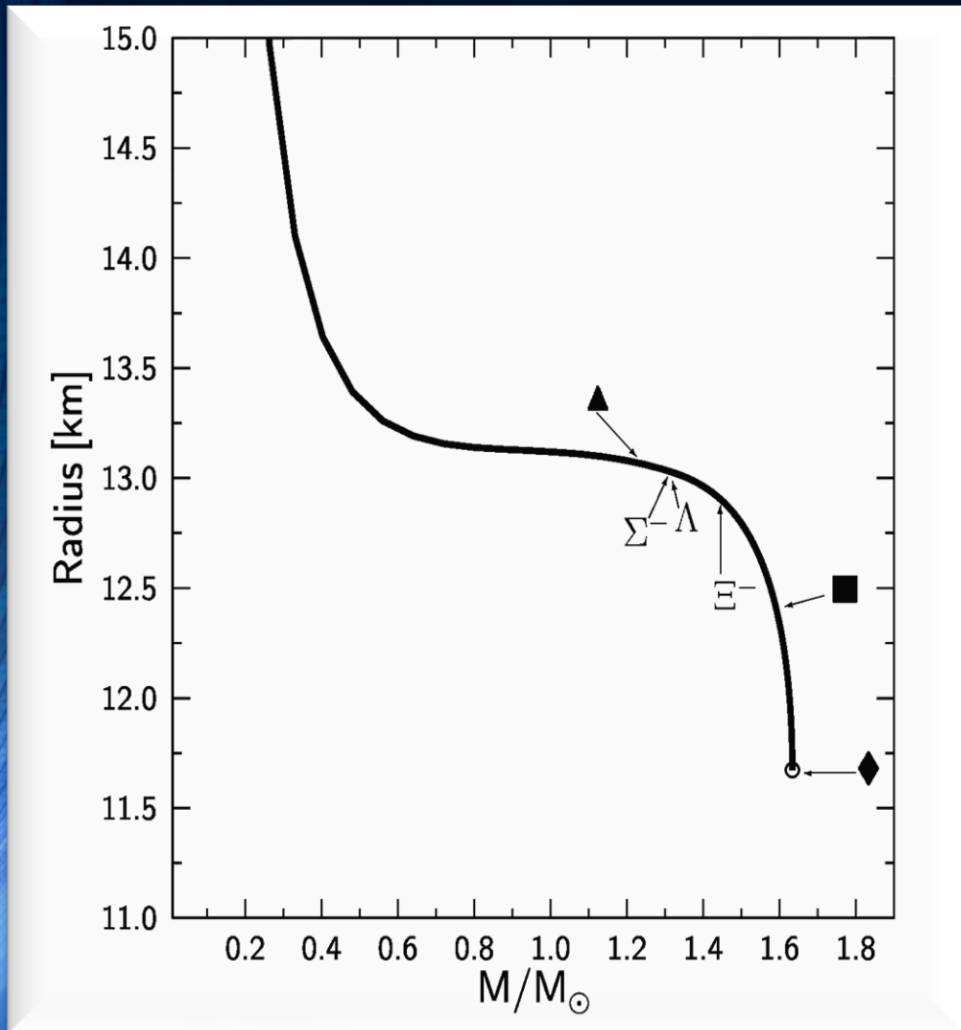
Masse, Energie und Impuls des Systems

Zustandsgleichung der Materie  
Druck ( Dichte , Temperatur )

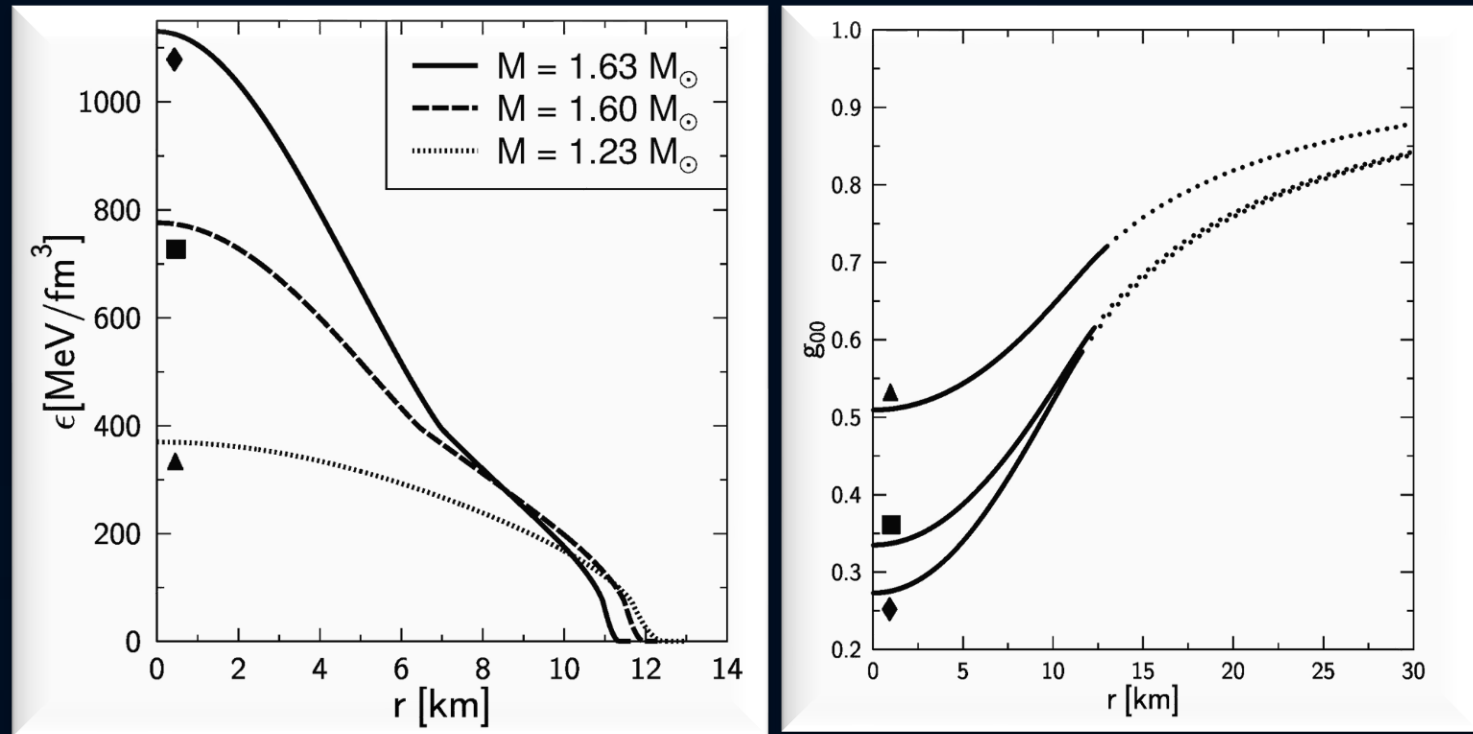


# Neutron Star Properties

Left: The neutron star radius as a function of its mass. A low, middle and high density star is displayed within the figure. Additionally the onset of hyperonic particles is visualized.



Middle: Energy density profiles of three neutron stars with different central densities and masses. The low density stars do not contain any hyperons, whereas the other two stars do have hyperons in their inner core.



Right: Time-time component of the metric tensor as a function of the radial coordinate. The solid line corresponds to the inner TOV-solution, whereas the dotted curve depicts the outer Schwarzschild part.

# Binäre Neutronenstern Systeme

Zurzeit kennt man ca. 25  
binäre Neutronenstern Systeme

Beispiel:

Der **Double Pulsar**

(PSR J0737-3039A/B):

Entdeckt im Jahre 2003

Eccentricity: 0.088

Pulsar A:  $P=23$  ms,  $M=1.3381(7)$

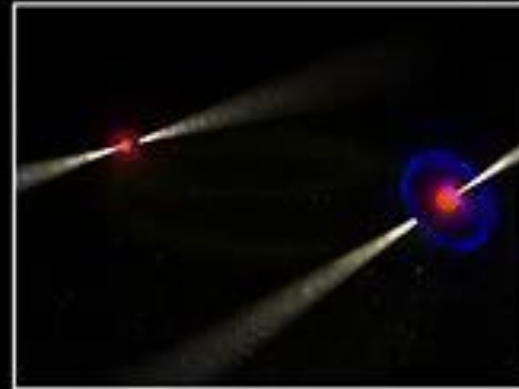
Pulsar B:  $P=2.7$  s,  $M=1.2489(7)$

Abstand zwischen den Sternen nur  
800,000 km

Orbitale Periode: 147 Minuten

Abstand verkleinert sich langsam  
aufgrund der Abstrahlung von  
Gravitationswellen

Die beiden Neutronensterne  
werden erst in 85 Millionen  
Jahren kollidieren



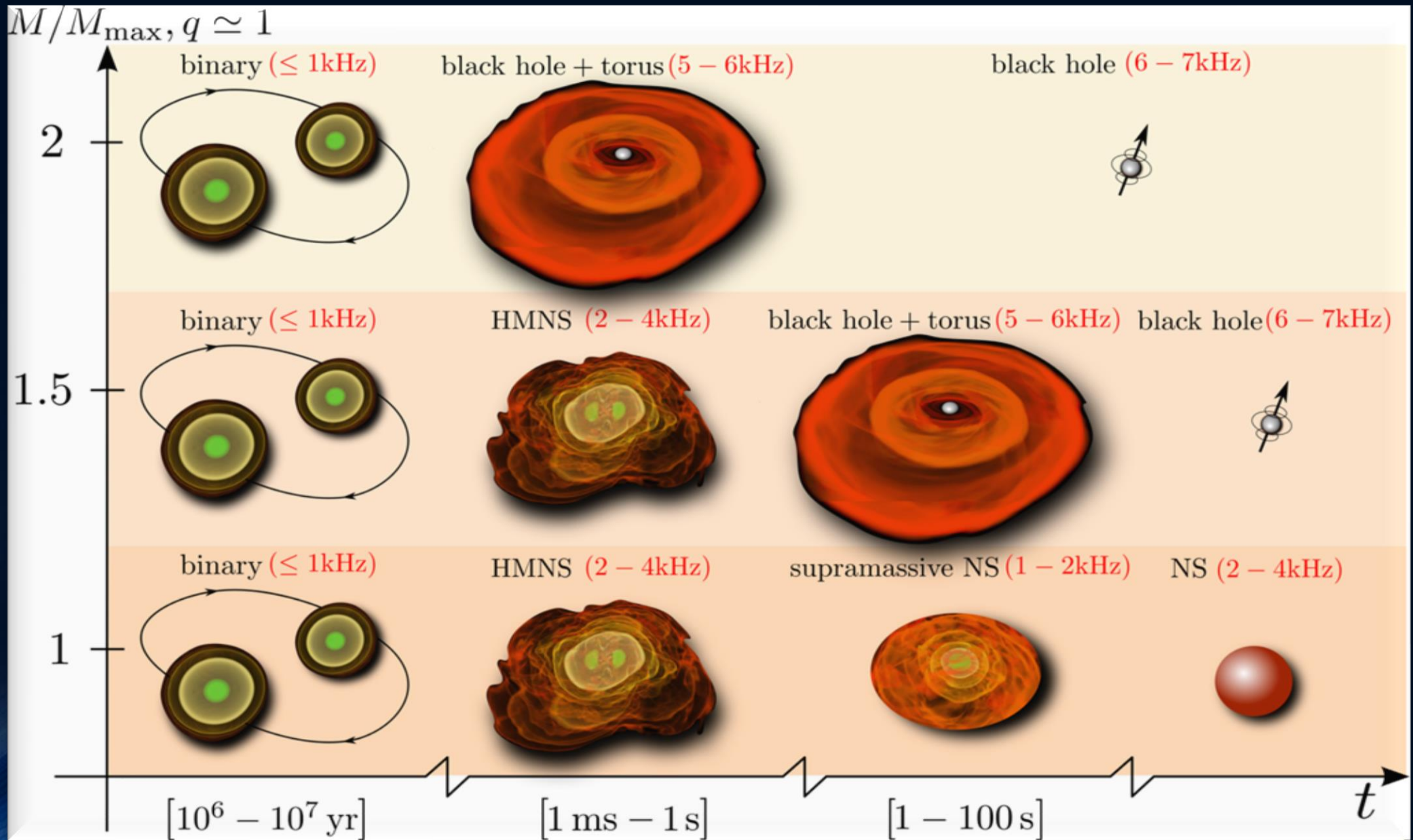


# Was geschieht wenn zwei Neutronensterne miteinander kollidieren?

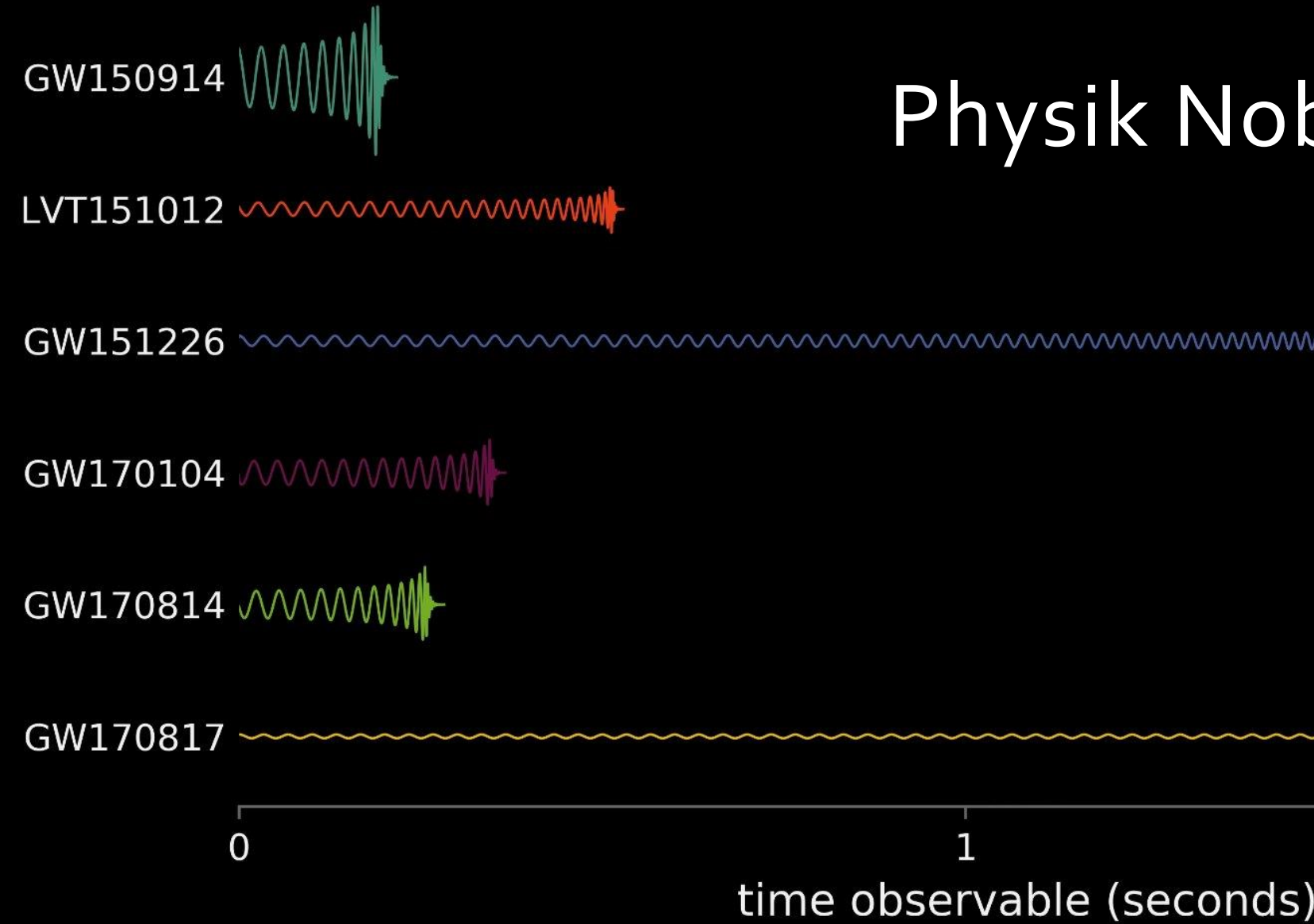
Zwei sehr massive Neutronensterne

Zwei mittelschwere Neutronensterne

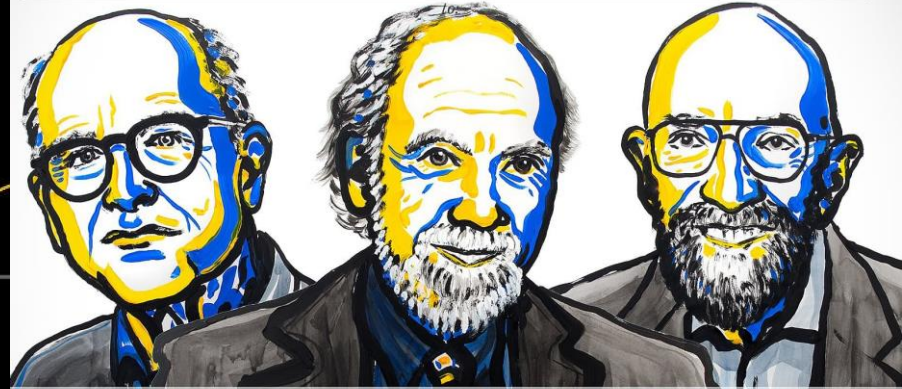
Zwei leichte Neutronensterne



# Physik Nobel Preis 2017



2017 NOBEL PRIZE IN PHYSICS



Rainer Weiss  
Barry C. Barish  
Kip S. Thorne



# Das lang ersehnte Ereignis GW170817

	Low-spin priors ( $ \chi  \leq 0.05$ )	High-spin priors ( $ \chi  \leq 0.89$ )
Primary mass $m_1$	1.36–1.60 $M_\odot$	1.36–2.26 $M_\odot$
Secondary mass $m_2$	1.17–1.36 $M_\odot$	0.86–1.36 $M_\odot$
Chirp mass $\mathcal{M}$	$1.188^{+0.004}_{-0.002} M_\odot$	$1.188^{+0.004}_{-0.002} M_\odot$
Mass ratio $m_2/m_1$	0.7–1.0	0.4–1.0
Total mass $m_{\text{tot}}$	$2.74^{+0.04}_{-0.01} M_\odot$	$2.82^{+0.47}_{-0.09} M_\odot$
Radiated energy $E_{\text{rad}}$	$> 0.025 M_\odot c^2$	$> 0.025 M_\odot c^2$
Luminosity distance $D_L$	$40^{+8}_{-14}$ Mpc	$40^{+8}_{-14}$ Mpc
Viewing angle $\Theta$	$\leq 55^\circ$	$\leq 56^\circ$
Using NGC 4993 location	$\leq 28^\circ$	$\leq 28^\circ$
Combined dimensionless tidal deformability $\tilde{\Lambda}$	$\leq 700$	$\leq 700$
Dimensionless tidal deformability $\Lambda(1.4M_\odot)$	$\leq 800$	$\leq 1400$

17. August 2017

Gravitationswelle einer  
Neutronenstern Kollision gemessen!

# Computer Simulation einer Neutronenstern Kollision

Credits: Cosima Breu, David Radice und Luciano Rezzolla



**Dichte der  
Neutronenstern Materie**

8.5 14



$\lg(\rho)$  [g/cm<sup>3</sup>]

**Temperatur der  
Neutronenstern Materie**

0 50



T [MeV]



# Allgemeine Relativitätstheorie mit dem Computer

## (General Theory of Relativity on the Computer)

Vorlesung gehalten an der

J.W.Goethe-Universität in Frankfurt am Main

(Sommersemester 2020)

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

Frankfurt am Main 23.05.2020

Python Jupiter Notebook  
(mit numerischer Lösung)

### Erster Vorlesungsteil: Allgemeine Relativitätstheorie mit Python

#### Die innere Schwarzschildlösung eines sphärisch symmetrischen, statischen Objektes (z.B. Erde, Neutronenstern)

#### Von der Einstein Gleichung zur Tolman-Oppenheimer-Volkoff Gleichung (TOV)

In den vorigen drei Vorlesungen wurde die Geodätengleichung in vorgegebener Schwarzschild Raumzeit für unterschiedliche Anfangsbedingungen numerisch analysiert. Die raumzeitliche Struktur, die Metrik, wurde hierbei als gegeben vorausgesetzt. In der folgenden Vorlesung betrachteten wir nun wie man die Metrik bei vorgegebener Materieverteilung berechnet. Die zugrundeliegende Gleichung die es hier zu lösen gilt ist die Einstein Gleichung.

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = 8\pi T_{\mu\nu}$$

Zunächst wird das Python Packet "GraviPy" eingebunden, welches auf dem Packet SymPy basiert und symbolische Berechnungen in der Allgemeinen Relativitätstheorie relativ einfach möglich macht.

Man erhält somit das folgende System von Differentialgleichungen, die so genannten "Tolman-Oppenheimer-Volkoff Gleichung (TOV Gleichungen)"

In [24]: Eq1

Out[24]: 
$$\frac{d}{dr}m(r) = 4\pi r^2 e(r)$$

In [25]: Eq2

Out[25]: 
$$\frac{d}{dr}\Phi(r) = \frac{2(4\pi r^3 p(r) + m(r))\Phi(r)}{r(r - 2m(r))}$$

In [26]: Eq5

Out[26]: 
$$\frac{d}{dr}p(r) = -\frac{4\pi r^3 e(r)p(r) + 4\pi r^3 p^2(r) + e(r)m(r) + m(r)p(r)}{r(r - 2m(r))}$$

## Numerische Lösung der TOV-Gleichungen

Im folgenden werden die TOV Gleichungen numerisch gelöst, indem wir einerseits eine Zustandsgleichung der Materie (eine Funktion  $p(e)$ , hier speziell  $p(e) = 10 e^{5/3}$ ) festlegen und von einem Startwert der zentralen Energiedichte im Inneren des sphärisch symmetrischen Objektes nach Außen integrieren.

In [27]: `TOV1=Eq1.subs(e, (p/10)**(3/5))`

In [28]: `TOV2=sym.simplify(Eq5.subs(e, (p/10)**(3/5)))`

In [29]: `TOV3=sym.simplify(Eq2.subs(e, (p/10)**(3/5)))`

## Lösen der TOV Gleichungen in Python

Das zugrundeliegende System von Differentialgleichungen lösen wir im folgenden mit der Funktion `odeint` aus dem SciPy-Paket

## Links

<http://docs.scipy.org/doc/>



## Links

<http://docs.scipy.org/doc/>

Beispiel Lösen von DGLs mit odeint <https://gertingold.github.io/eidprog/scipy.html>

Tutorial sympy [https://www.grund-wissen.de/informatik/python/\\_downloads/tutorial-scipy-sympy.pdf](https://www.grund-wissen.de/informatik/python/_downloads/tutorial-scipy-sympy.pdf)

```
In [30]: from scipy import integrate
#help(integrate.odeint)
```

Wir schreiben jedoch zunächst das System der TOV-Gleichungen ein wenig um und benennen  $m(r) = y_1$ ,  $p(r) = y_2$  und  $\Phi(r) = y_3$

```
In [31]: y1, y2, y3 = symbols('y1, y2, y3')
TOV1a=TOV1.subs([(Fm,y1),(p,y2),(Spi,math.pi)])
TOV2a=TOV2.subs([(Fm,y1),(p,y2),(Spi,math.pi)])
TOV3a=TOV3.subs([(Fm,y1),(p,y2),(Fphi,y3),(Spi,math.pi)])
```

```
In [32]: TOV1a
```

```
Out[32]:  $\frac{d}{dr}y_1 = 3.15652958395295r^2y_2^{0.6}$ 
```

```
In [33]: TOV2a
```

```
Out[33]:  $\frac{d}{dr}y_2 = -\frac{12.5663706143592r^3y_2^2 + 3.15652958395295r^3y_2^{1.6} + 0.251188643150958y_1y_2^{0.6} + y_1y_2}{r(r - 2y_1)}$ 
```

```
In [34]: TOV3a
```

```
Out[34]:  $\frac{d}{dr}y_3 = \frac{2y_3(12.5663706143592r^3y_2 + y_1)}{r(r - 2y_1)}$ 
```

Zusätzlich benutzen wir die Funktion lambdify() um die numerischen Werte der TOV Gleichungen zu erhalten (siehe <https://docs.sympy.org/latest/modules/utilities/lambdify.html>):

Zusätzlich benutzen wir die Funktion `lambdify()` um die numerischen Werte der TOV Gleichungen zu erhalten (siehe <https://docs.sympy.org/latest/modules/utilities/lambdify.html>):

```
In [35]: TOV1b=sym.lambdify([r,(y1,y2,y3)], TOV1a.rhs)
```

```
In [36]: TOV2b=sym.lambdify([r,(y1,y2,y3)], TOV2a.rhs)
```

```
In [37]: TOV3b=sym.lambdify([r,(y1,y2,y3)], TOV3a.rhs)
```

Wir definieren das Differentialgleichungssystem der TOV Gleichungen als eine Funktion

```
In [38]: def DGLSysTOV(y, r):
          y1, y2, y3 = y
          dy1 = TOV1b(r, (y1,y2,y3))
          dy2 = TOV2b(r, (y1,y2,y3))
          dy3 = TOV3b(r, (y1,y2,y3))
          return np.array([dy1, dy2, dy3])
```

Zur numerischen Lösung (mittels der Funktion `odeint()`) müssen wir den eindimensionalen Raum der radialen Koordinate  $r$  in einzelne Stützstellen aufteilen (hier speziell  $r \in [10^{-14}, 30]$  mit 10000 Punkten). Da die TOV Gleichungen bei  $r = 0$  singular sind, beginnen wir die Integration einfach bei einem sehr kleinen Wert nahe Null ( $10^{-14}$ ).

```
In [80]: pts = np.linspace(10**(-14), 30, 10001)
```

Als Rand- bzw. Anfangsbedingungen der Integration setzen wir die integrierte Masse im Sternzentrum auf Null ( $m(r = 10^{-14}) = 0$ ), den zentralen Druck auf  $p(r = 10^{-14}) = 10 (0.0005)^{5/3}$  und die Metrikkomponente  $\Phi$  willkürlich zunächst auf Eins ( $\Phi(r = 10^{-14}) = 1$ ).

```
In [81]: Randbedingungen = np.array([0, 10*(0.0005)**(5/3), 1])
```

Das eigentliche Lösen der TOV Gleichung:

```
In [82]: LoesTOV = integrate.odeint(DGLSysTOV, Randbedingungen, pts)
```

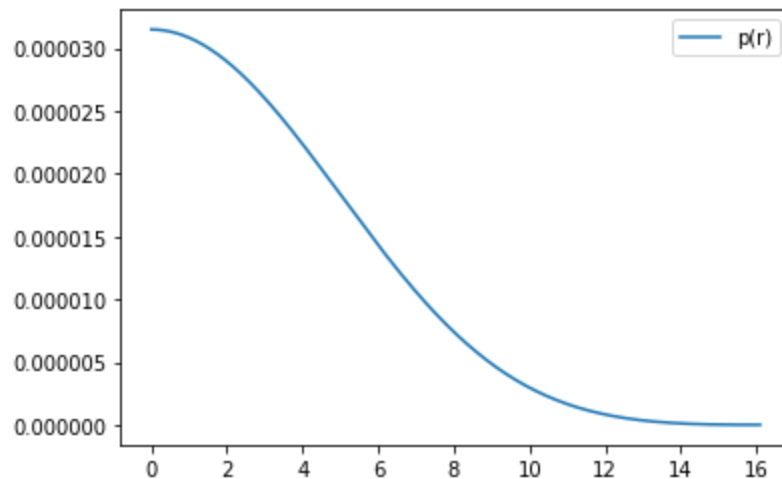


Das eigentliche Lösen der TOV Gleichung:

```
In [82]: LoesTOV = integrate.odeint(DGLSysTOV, Randbedingungen, pts)
<string>:3: RuntimeWarning: invalid value encountered in double_scalars
<string>:3: RuntimeWarning: invalid value encountered in double_scalars
```

Wir können uns nun z.B. die Druckverteilung  $p(r)$  im Sterninneren

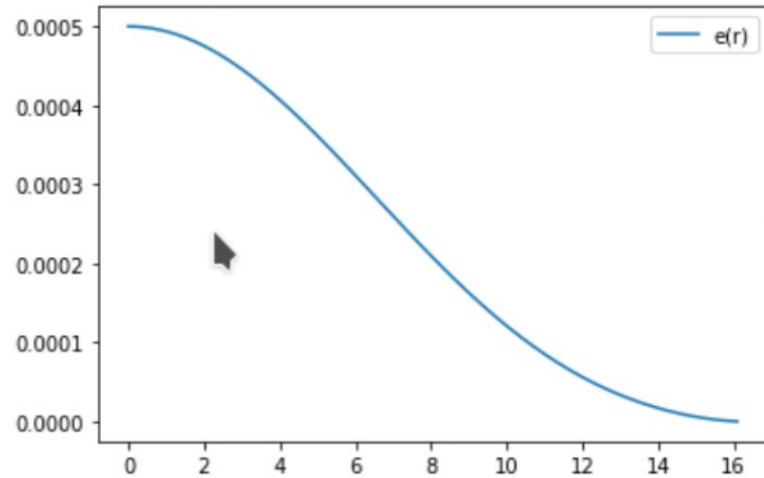
```
In [83]: plt.clf()
fig, ax = plt.subplots(num=1)
ax.plot(pts, LoesTOV[:, 1], '-', label='p(r)')
ax.legend(loc='best')
plt.show()
```



, die Energiedichteverteilung  $e(r)$

```
In [84]: plt.clf()
fig, ax = plt.subplots(num=1)
ax.plot(pts, (LoesTOV[:, 1]/10)**(3/5), '-', label='e(r)')
ax.legend(loc='best')
plt.show()
```

```
In [84]: plt.clf()
fig, ax = plt.subplots(num=1)
ax.plot(pts, (LoesTOV[:, 1]/10)**(3/5), '-', label='e(r)')
ax.legend(loc='best')
plt.show()
```



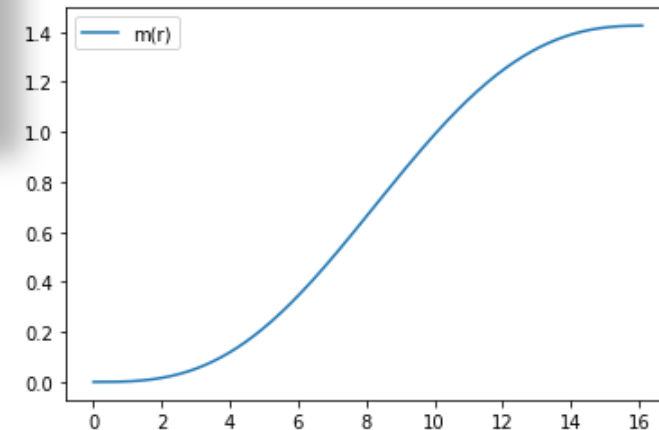
, oder die Sternmasse  $m(r)$  als Funktion des radialen Abstands  $r$  vom Sternzentrum betrachten.

```
In [85]: plt.clf()
fig, ax = plt.subplots(num=1)
ax.plot(pts, LoesTOV[:, 0], '-', label='m(r)')
ax.legend(loc='best')
plt.show()
```

## Energiedichten und Masse Profile innerhalb des Neutronensterns

, oder die Sternmasse  $m(r)$  als Funktion des radialen Abstands  $r$  vom Sternzentrum betrachten.

```
plt.clf()
fig, ax = plt.subplots(num=1)
ax.plot(pts, LoesTOV[:, 0], '-', label='m(r)')
ax.legend(loc='best')
plt.show()
```





Aufgrund des Birkov Theorems muss die Innenlösung der Metrik in die äussere Schwarzschildmetrik am Sternrand übergehen. Da wir nun die Gesamtmasse und den Radius des Sterns kennen, können wir auch die innere  $g_{tt}$  und  $g_{rr}$  Komponente der Metrik angeben. Der zuvor gewählte Randwert der Metrikkomponente  $\Phi$  ( $\Phi(r = 10^{-14}) = 1$ ) war willkürlich gewählt und erst nachdem die Gesamtmasse und den Radius des Sterns durch die numerischen Lösung bekannt sind, kann man die wirkliche Form der  $g_{tt}$ -Komponente ( $g_{tt}(r) = \Phi(r)$ ) durch eine y-Achsenverschiebung erhalten. Der Radius des Sterns beträgt (in km)

```
In [86]: j=0
while math.isnan(LoesTOV[j, 0])==False:
    j=j+1
i=j-1
R=pts[i]
R
```

Out[86]: 16.095000000000002

, seine Masse (in km)

```
In [87]: M=LoesTOV[i, 0]
M
```

Out[87]: 1.4250357645821545

und seine  $g_{tt}$ -Komponente am Sternrand sollte dem Wert der Schwarzschildmetrik entsprechen:  $g_{tt}(R) = \Phi(R) = (1 - \frac{2M}{R}) =$

```
In [88]: Phi0=(1-2*M/R)
Phi0
```

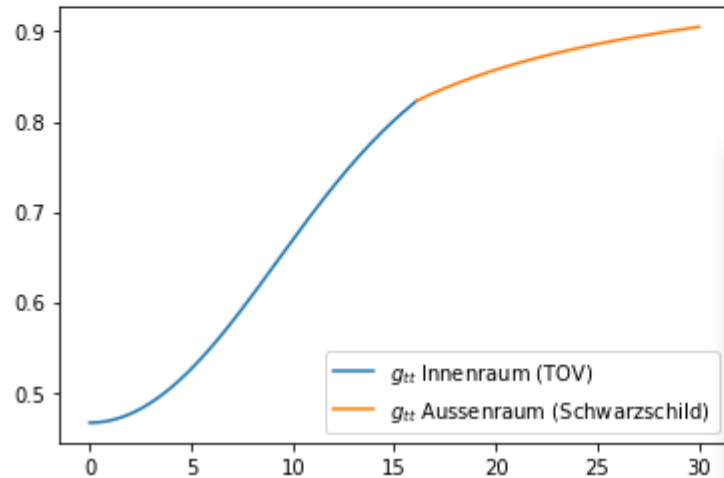
Out[88]: 0.8229219304650942

```
In [89]: abweichphi=LoesTOV[i, 2]-Phi0
```

Veranschaulichung der  $g_{tt}$ -Komponente der Innenraummetrik und Außenraummetrik.

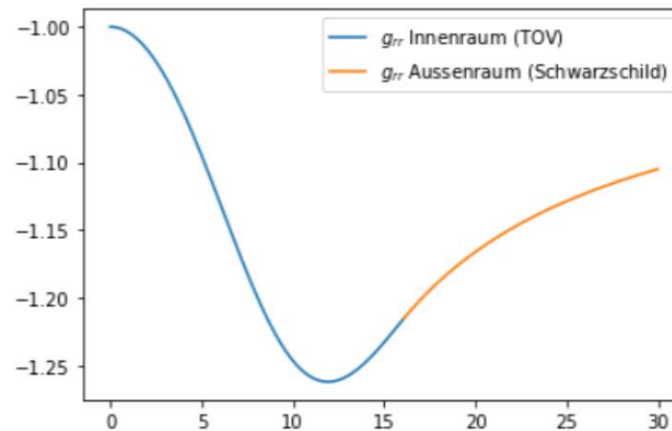
Veranschaulichung der  $g_{tt}$ -Komponente der Innenraummetrik und Außenraummetrik.

```
In [90]: plt.clf()
fig, ax = plt.subplots(num=1)
ax.plot(pts, LoesTOV[:, 2]-abweichphi, '-', label='$g_{tt}$ Innenraum (TOV)')
ax.plot(pts[i:], (1-2*M/pts[i:]), '-', label='$g_{tt}$ Aussenraum (Schwarzschild)')
ax.legend(loc='lower right')
plt.show()
```



Metrik Komponenten  
tt: links  
rr: unten  
innerhalb (blau) und  
außerhalb (orange) des  
Neutronensterns

```
In [91]: plt.clf()
fig, ax = plt.subplots(num=1)
ax.plot(pts, -(1-2*LoesTOV[:, 0]/pts[:])**(-1), '-', label='$g_{rr}$ Innenraum (TOV)')
ax.plot(pts[i:], -(1-2*M/pts[i:])**(-1), '-', label='$g_{rr}$ Aussenraum (Schwarzschild)')
ax.legend(loc='upper right')
plt.show()
```



Im folgenden wird eine Sequenz von Sternen mit unterschiedlichen zentralen Energiedichten-Werten berechnet. Trägt man die Gesamtmasse der einzelnen Sterne gegen deren Radius auf, so erhält man die Masse-Radius Beziehung. Jede Zustandsgleichung besitzt eine eigene Obergrenze, die sogenannte maximale Masse. Die Berechnung erfolgt durch eine for-Schleife über einen geeigneten zentralen Energiedichtebereich:

```
In [92]: pts = np.linspace(10**(-14), 30, 10001)
Liste0=[]
ListM=[]
ListR=[]
for e0 in np.linspace(0.0002, 0.004, 40):
    Randbedingungen = np.array([0, 10*(e0)**(5/3), 1])
    LoestTOV = integrate.odeint(DGLSysTOV, Randbedingungen, pts)
    j=0
    while math.isnan(LoestTOV[j, 0])==False:
        j=j+1
    i=j-1
    Liste0.append(e0)
    ListR.append(pts[i])
    ListM.append(LoestTOV[i, 0])
```

```
<string>:3: RuntimeWarning: invalid value encountered in double_scalars
<string>:3: RuntimeWarning: invalid value encountered in double_scalars
```

Die berechnete Masse-Radius Beziehung können wir uns in einem R-M Diagramm darstellen:

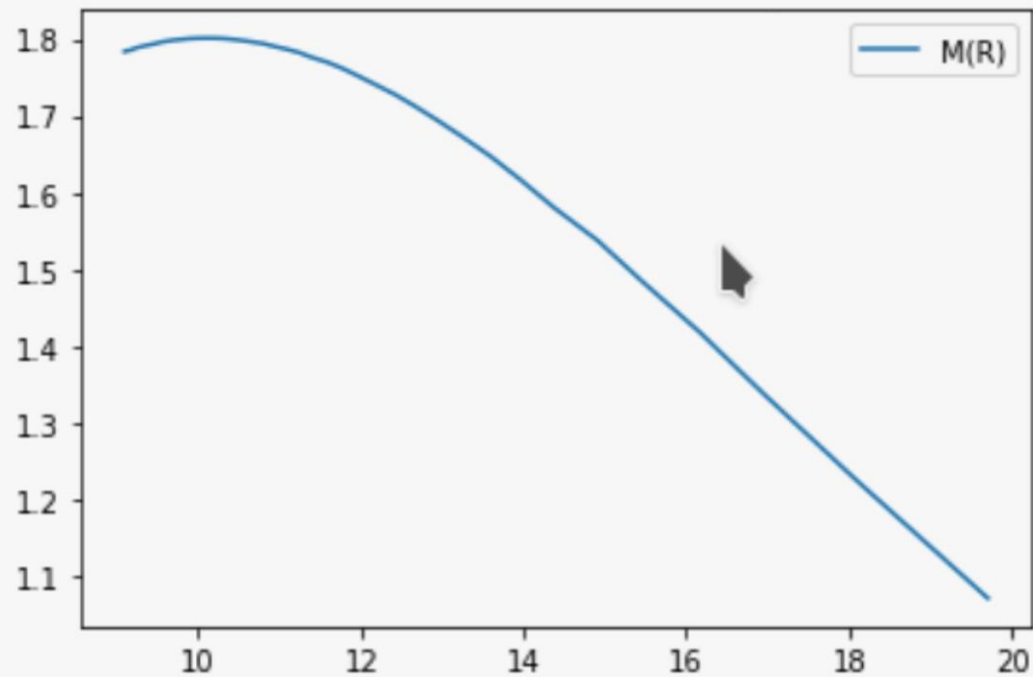
```
In [79]: plt.clf()
fig, ax = plt.subplots(num=1)
ax.plot(ListR, ListM, '-', label='M(R)')
ax.legend(loc='best')
plt.show()
```



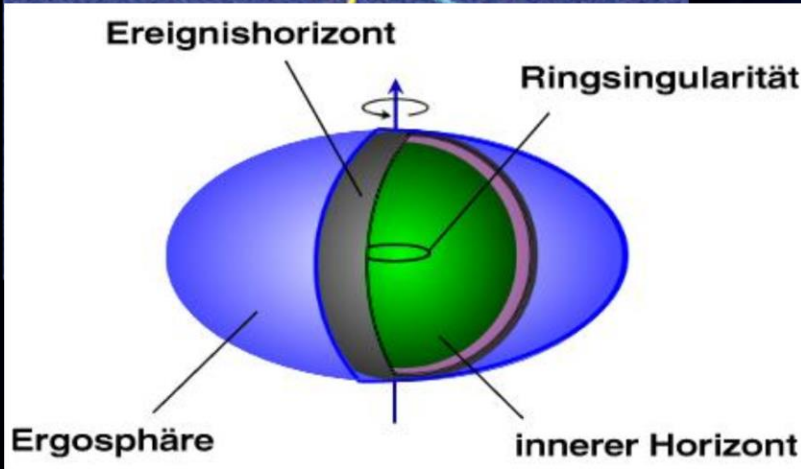
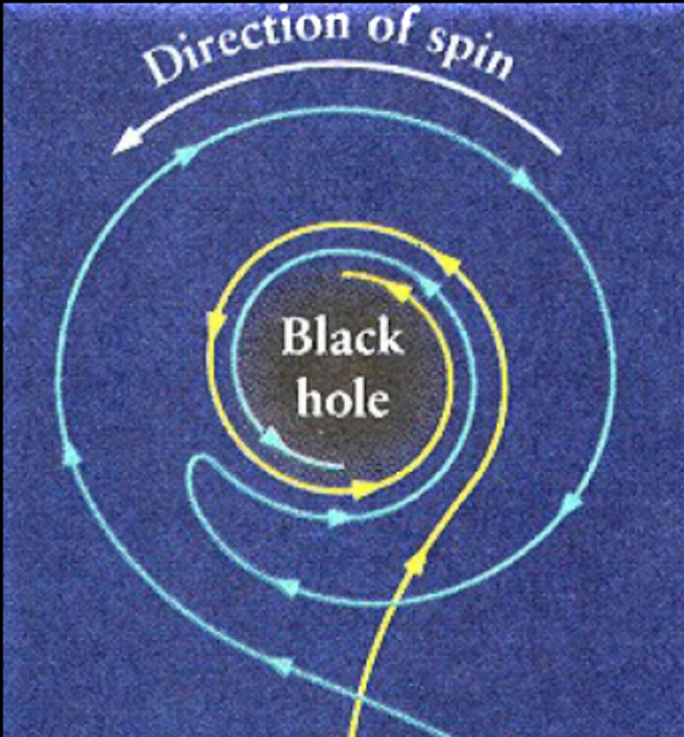
# Masse-Radius Relation bei gewählter Zustandsgleichung

Die berechnete Masse-Radius Beziehung können wir uns in einem R-M Diagramm darstellen:

```
In [79]: plt.clf()  
fig, ax = plt.subplots(num=1)  
ax.plot(ListR, ListM, '-', label='M(R)')  
ax.legend(loc='best')  
plt.show()
```



# Rotierende schwarze Löcher und die Kerr Metrik

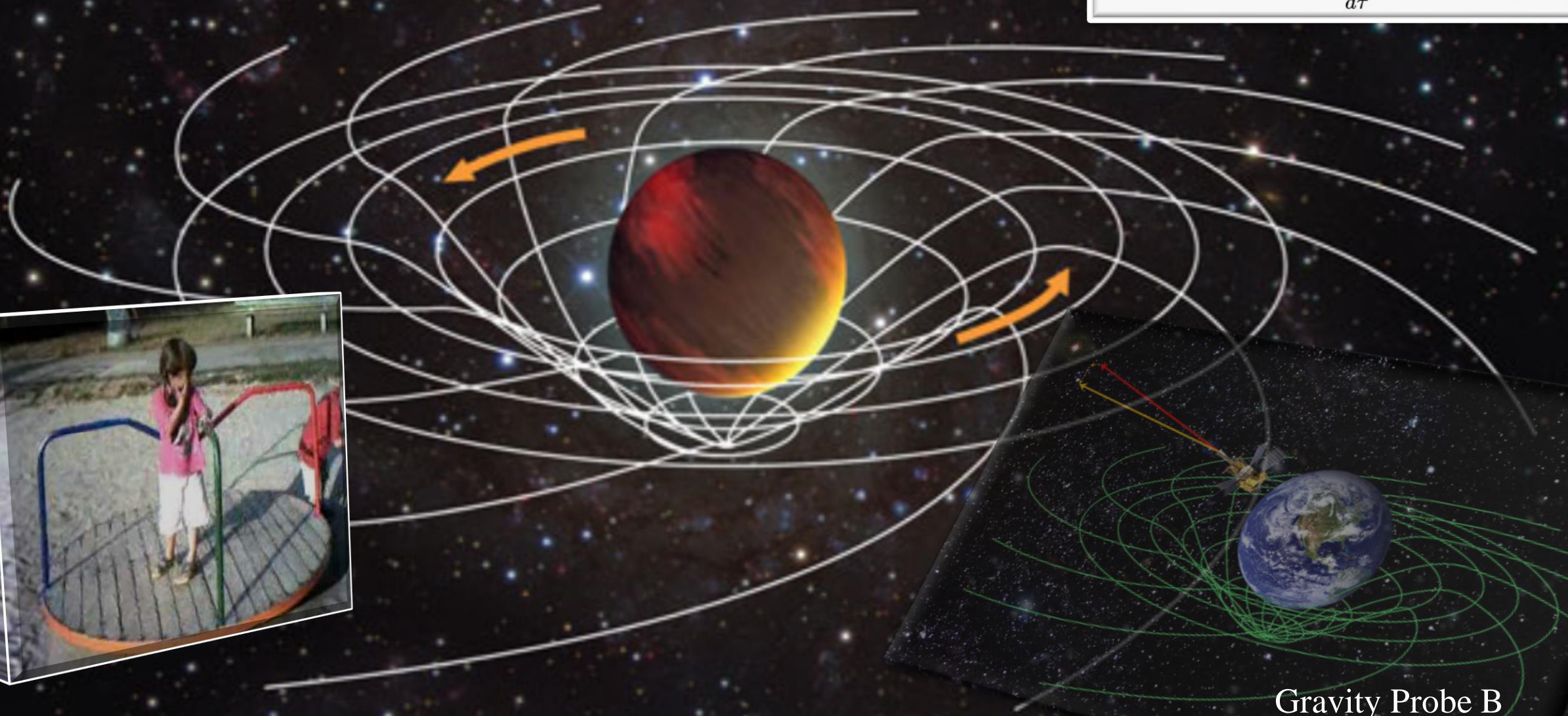




# Der Mitführungseffekt der Raumzeit ("Frame-Dragging")

Experimente zur Bestätigung des Effektes: LARES, Gravity Probe B

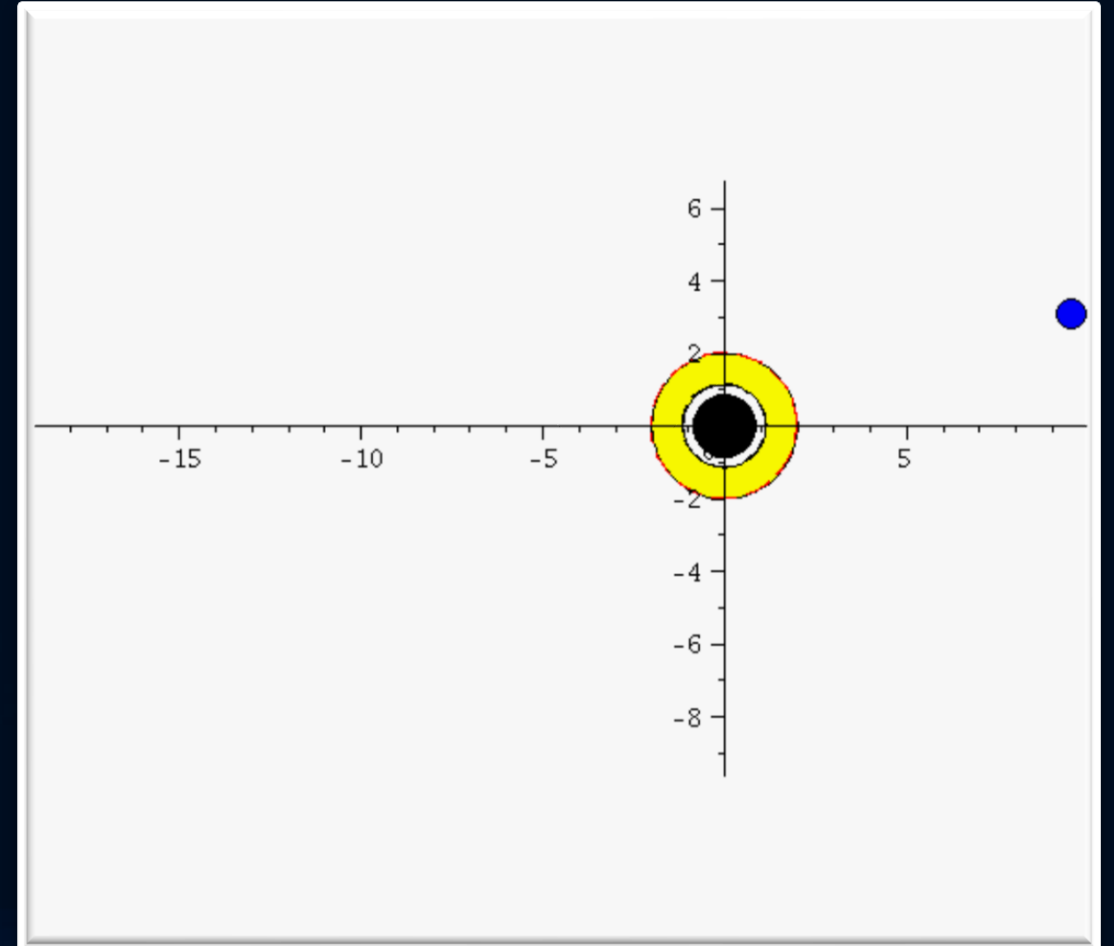
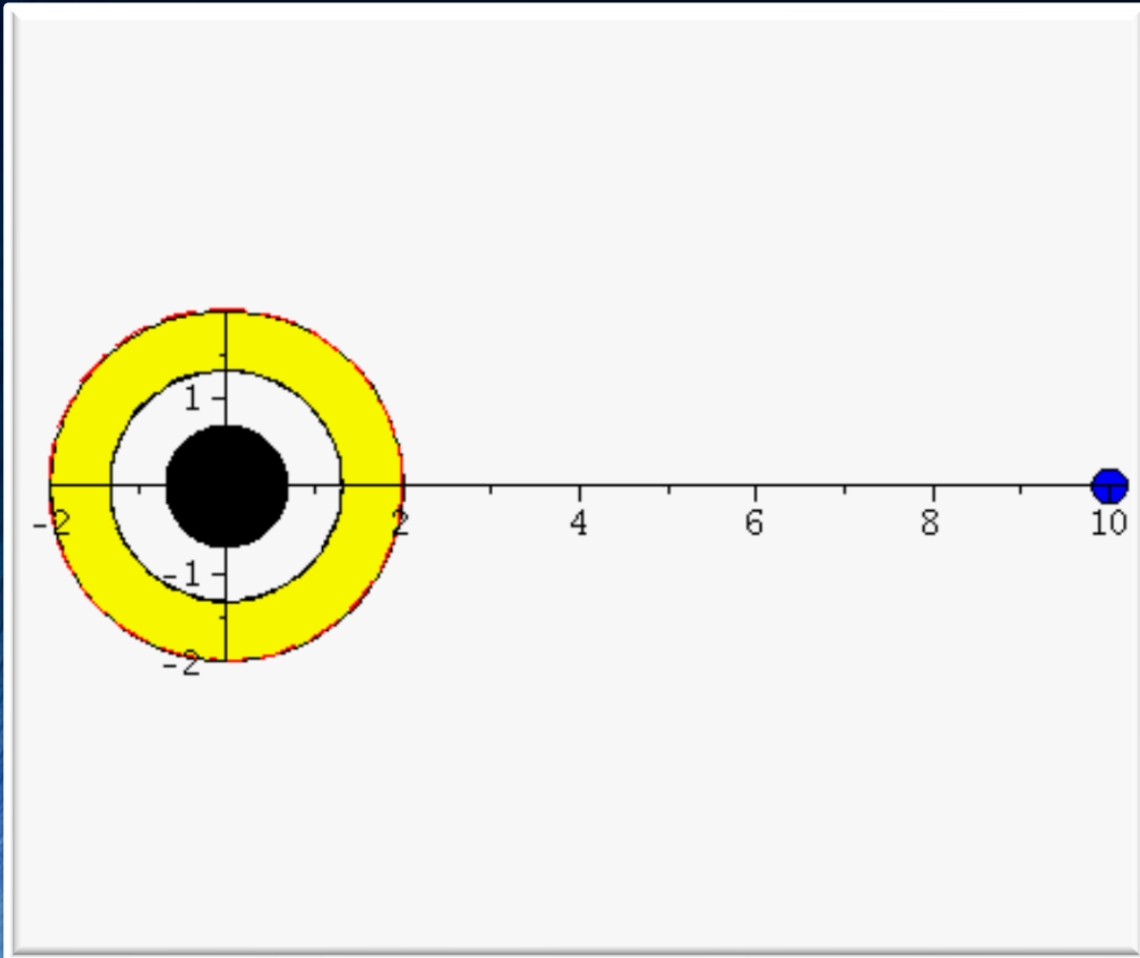
$$\omega(r, \theta) = \frac{d\phi}{dt} = \frac{\frac{d\phi}{d\tau}}{\frac{dt}{d\tau}} = \frac{u^\phi}{u^t} = \frac{g^{t\phi}}{g^{tt}}$$



Gravity Probe B



# Der Mitführungseffekt der Raumzeit ("Frame-Dragging") und der Gravitomagnetische Effekt

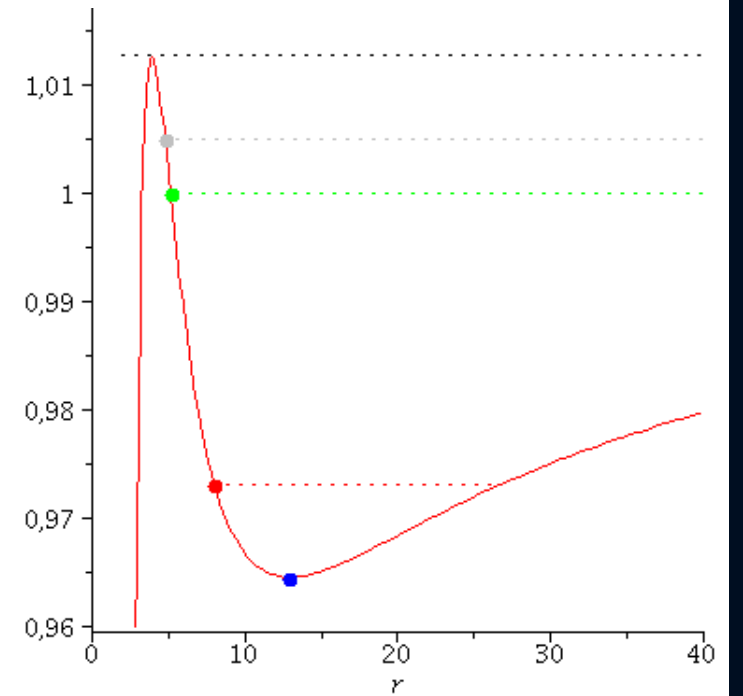
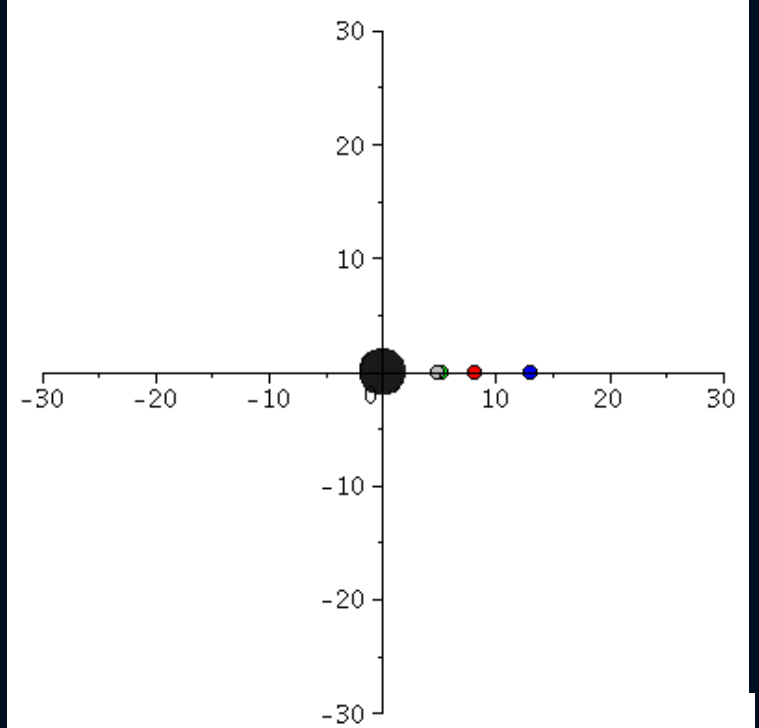
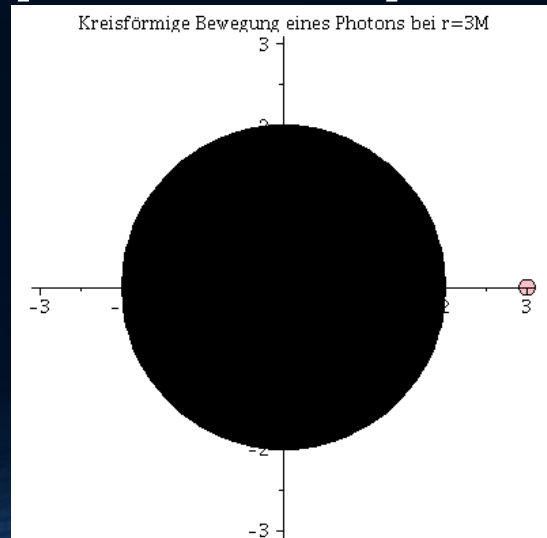
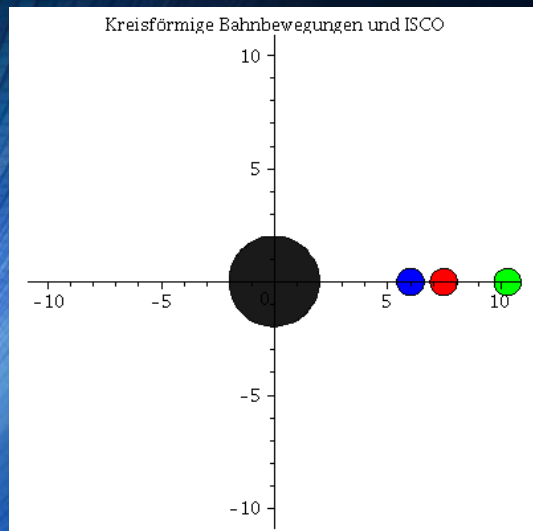


# Geodätische Bewegung um ein nicht-rotierendes schwarzes Loch

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = -8\pi T_{\mu\nu}$$

$$\frac{d^2 x^\mu}{d\tau^2} + \Gamma^\mu_{\nu\rho} \frac{dx^\nu}{d\tau} \frac{dx^\rho}{d\tau} = 0$$

## The *ISCO* and the *photon sphere*



[Einführung](#)

[Teil I: Analytische Berechnungen und numerische Simulationen in Maple](#)

[Teil II: Paralleles Programmieren mit C++ und OpenMP/MPI](#)

[Teil III: Computersimulationen mit dem Einstein-Toolkit](#)

[Aufgaben](#)

[Download Maple Worksheed](#)

# Allgemeine Relativitätstheorie mit dem Computer

## General Theory of Relativity on the Computer

Vorlesung gehalten an der J.W.Goethe-Universität in Frankfurt am Main (Sommersemester 2016)

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

Frankfurt am Main 11.04.2016

[Erster Vorlesungsteil: Allgemeine Relativitätstheorie mit Maple](#)

[5. Vorlesung](#)

### **Einführung**

Basierend auf den Ergebnissen der dritten Vorlesung des ersten Teils wird im folgenden die Bewegung eines Probekörpers um ein rotierendes schwarzes Loch vorgestellt.

### **Bewegung eines Probekörpers um ein rotierendes schwarzes Loch**

Im folgenden wird die Geodätengleichung in vorgegebener Kerr-Raumzeit (in Boyer-Lindquist Koordinaten) betrachtet. Die Geodätengleichung beschreibt wie sich ein Probekörper (Masse = 0) im Raum bewegt und sagt voraus, dass diese Bewegung sich stets entlang der kürzesten Kurve, in der durch die Metrik beschriebenen gekrümmten



# Eigenschaften der Kerr-Metrik

> **restart:**  
**with( tensor ):**  
**with(plots):**  
**with(plottools):**

Definition der kovarianten Raumzeit-Metrik eines rotierenden schwarzen Lochs der Masse M und Rotation a in Boyer-Lindquist Koordinaten (a ist ein spezifischer Drehimpuls a=J/M und wird als der sogenannte Kerr-Rotationsparameter bezeichnet). Die Kerr Metrik besitzt folgendes Aussehen:

$$g_{\mu\nu} = \begin{pmatrix} g_{tt}(r, \theta) & 0 & 0 & g_{t\phi}(r, \theta) \\ 0 & g_{rr}(r, \theta) & 0 & 0 \\ 0 & 0 & g_{\theta\theta}(r, \theta) & 0 \\ g_{\phi t}(r, \theta) & 0 & 0 & g_{\phi\phi}(r, \theta) \end{pmatrix}, \text{ wobei:}$$
$$g_{tt}(r, \theta) = \left( \frac{1 - 2Mr}{\rho^2} \right), \quad g_{t\phi}(r, \theta) = \frac{2aMr \sin^2(\theta)}{\rho^2}, \quad g_{rr}(r, \theta) = -\frac{\rho^2}{\Delta},$$
$$g_{\theta\theta}(r, \theta) = -\rho^2, \quad g_{\phi\phi}(r, \theta) = -\left( \frac{r^2 + a^2 + 2Mra^2 \sin^2(\theta)}{\rho^2} \right) \sin^2(\theta),$$
$$\rho^2 = r^2 + a^2 \cos^2(\theta), \quad \Delta = r^2 - 2Mr + a^2$$

## Struktur der Ereignishorizonte, Flächen der stationären Grenze und Flächen unendlicher Rotverschiebung

Die Flächen der stationären Grenze (stationary limit surfaces) und die der unendlichen Rotverschiebung sind durch  $g_{tt} = 0$  bestimmt. Man erhält zwei Lösungen, die man gewöhnlich mit den Symbolen  $r_{S^+}$  und  $r_{S^-}$  bezeichnet.

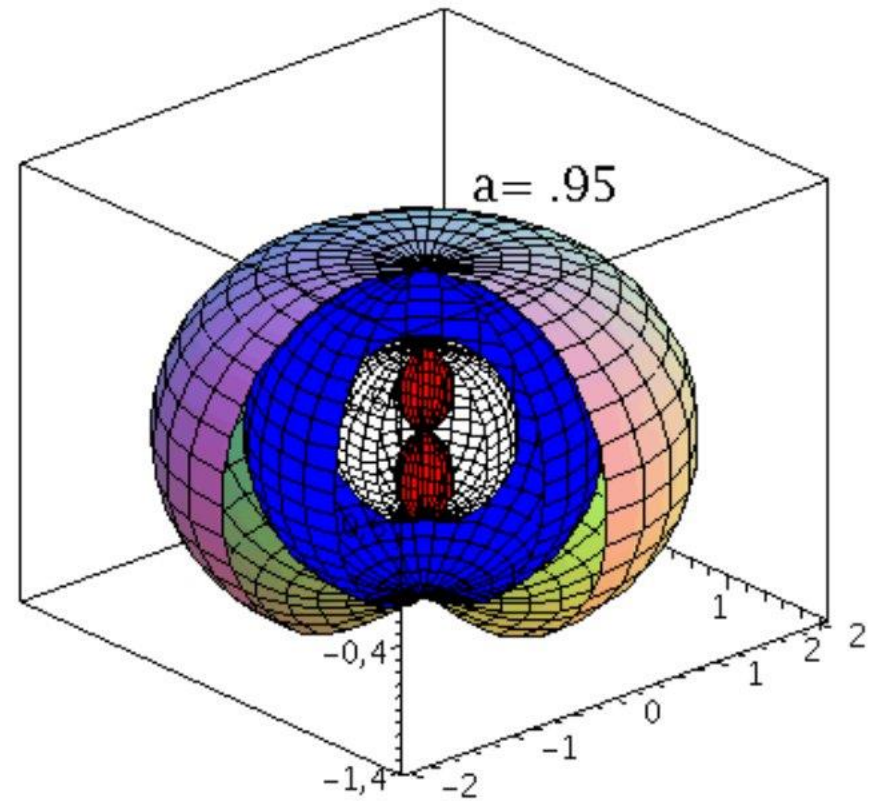
```
> UnRot:=solve(get_compts(g)[1,1]=0,r);
```

$$\text{UnRot} := M + \sqrt{M^2 - a^2 \cos^2(\theta)}, M - \sqrt{M^2 - a^2 \cos^2(\theta)} \quad (2.1.1.1)$$

Die Ereignishorizonte sind durch  $g^{rr} = 0$  (bzw.  $g_{rr} \rightarrow \infty$ ) bestimmt. Man erhält wieder zwei Lösungen, die man gewöhnlich mit den Symbolen  $r_+$  und  $r_-$  bezeichnet.

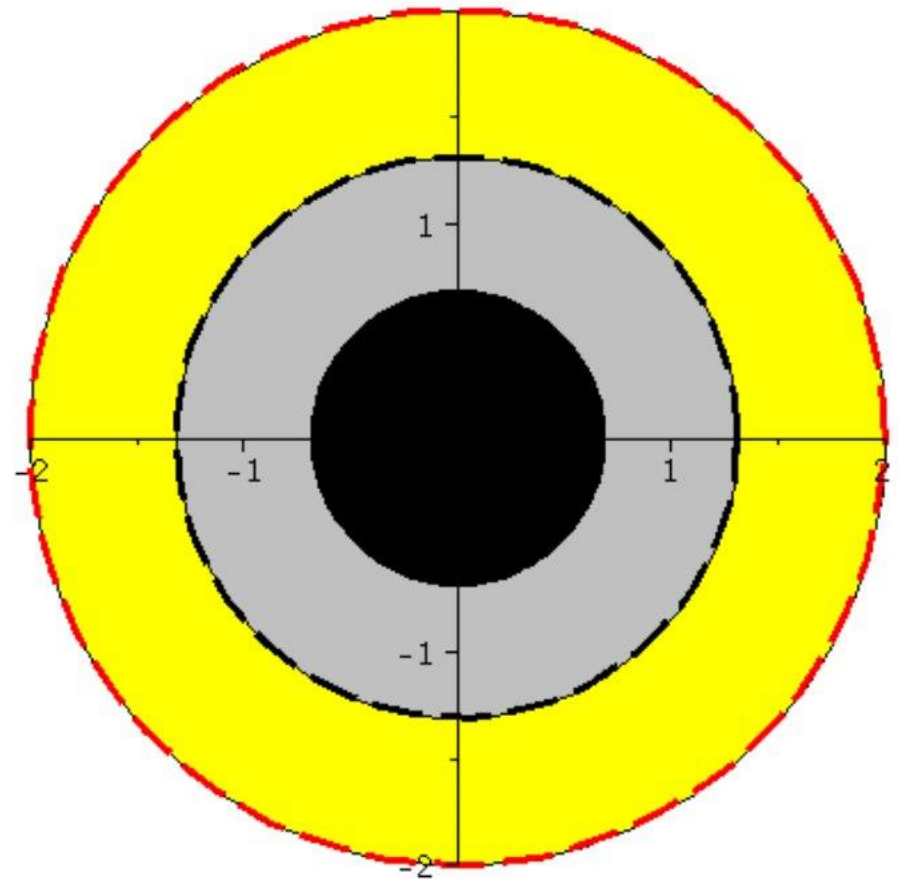
```
> Horizon:=solve(get_compts(ginv)[2,2]=0,r);
```

$$\text{Horizon} := M + \sqrt{M^2 - a^2}, M - \sqrt{M^2 - a^2} \quad (2.1.1.2)$$



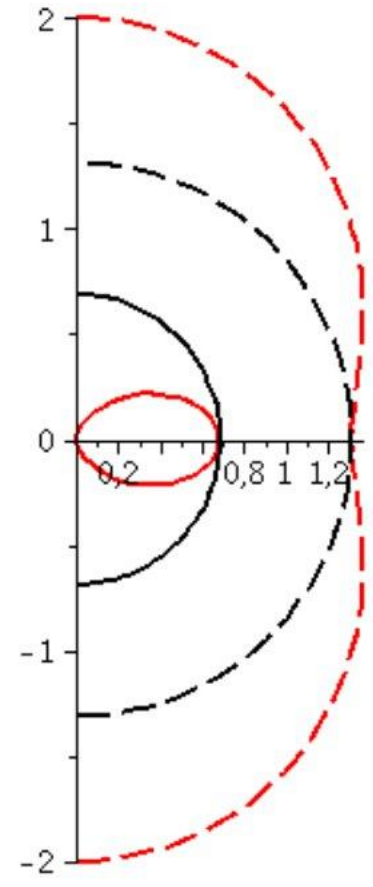
Horizontstruktur in der äquatorialen Ebene (Gelb: Ergosphäre, grau: Bereich zwischen äußerem und innerem Ereignishorizont,  $a=0.95$ ):





Horizontstruktur in der polaren Ebene ( $a=0.95$ ):

```
display({A,B,C,DD}, scaling=constrained);
```



Animation der Horizontstruktur bei ansteigender Rotation des schwarzen Lochs:

1	-3	1	-3	1	-3
---	----	---	----	---	----

## ***Die Rotation der raumzeitlichen Struktur um das schwarze Loch (das Frame dragging)***

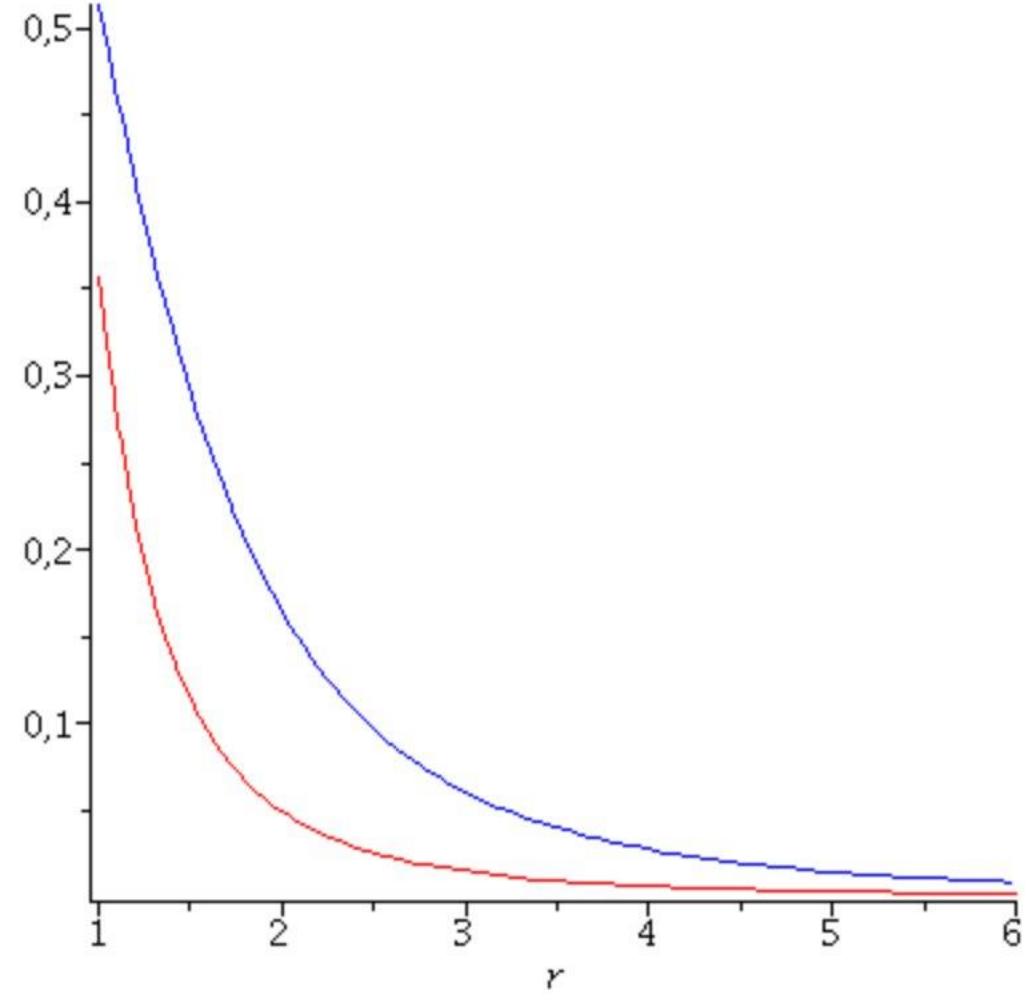
Ein rotierendes schwarzes Loch zieht die Raumzeit mit sich mit. Die Rotationsfrequenz mit der die Raumzeit mitgeführt wird nennt man "Frame dragging" Frequenz; sie quantisiert, mit welcher Frequenz ein im Eigensystem "ruhender" Probekörper von der ihm zugrundeliegenden Raumzeit mitgezogen wird:

$$\omega(r) = \frac{d\phi}{dt} = \frac{\frac{d\phi}{d\tau}}{\frac{dt}{d\tau}} = \frac{u^\phi}{u^t} = \frac{g^{t\phi}}{g^{tt}}$$

```
> FrameDrag:=get_compts(ginv)[1,4]/get_compts(ginv)[1,1];
#FrameDrag:=-get_compts(g)[1,4]/get_compts(g)[4,4];
```



Frame dragging Frequenz ( $a=0.95$  (blau) ,  $a=0.2$  (rot) )



# Radial in ein rotierendes schwarzes Loch einfallender Probekörper

Wir betrachten nun einen Probekörper der radial in ein rotierendes schwarzes Loch fällt.

```
> restart:
with( tensor ):
with(plots):
with(plottools):
```

Definition der kovarianten Raumzeit-Metrik eines rotierenden schwarzen Lochs der Masse  $M$  und Rotation  $a$  in Kerrschildkoordinaten:

```
> coord := [t, r, theta, phi]:
rho2:=r^2+(a*cos(theta))^2:
Delta:=r^2-2*M*r+a^2:Sig2:=(r^2+a^2)^2-a^2*Delta*(sin(theta))^2:
g_compts := array(symmetric,sparse, 1..4, 1..4):
g_compts[1,1] := (1-2*M*r/rho2):
g_compts[1,4] := +(2*a*M*r*(sin(theta))^2)/rho2:
g_compts[2,2] :=-rho2/Delta:
g_compts[3,3] := -rho2:
```

```
g_compts[4,4] := (r^2+a^2-2*M*r*a^2*(sin(theta))^2)/rho2:
```

## Anfangswerte:

Zur Zeit  $t=0$  sei der fallende Körper an der folgenden Position:  $(r=10, \theta=\pi/2, \phi=0)$ , die Anfangsgeschwindigkeit des Körpers sei 0. Wir beschreiben den Fall aus der Sichtweise eines im Unendlichen ruhenden Beobachters. Bemerkung: Der Anfangswert  $dt_0$  ergibt sich hierbei aus dem infinitesimalen Weglängenelement  $ds^2=1$  eines massiven Probekörpers:

```
> dt=solve(evalf(subs({M=setM,a=seta,theta=Pi/2,dr=0,dtheta=0,dphi=0,r=10},ds2=1)),dt);
```

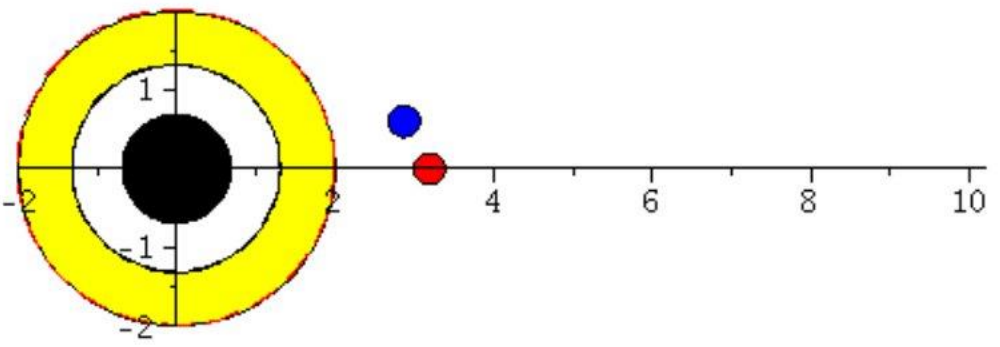
$$dt = (1.118033989, -1.118033989) \quad (2.2.4)$$

```
>      r0:=10:
      t0:=0:
      theta0:=Pi/2:
      phi0:=0:
      dr0:=0:
      dt0:=evalf(1/sqrt(1-2/r0)):
      dtheta0:=0:
      dphi0:=0:
```

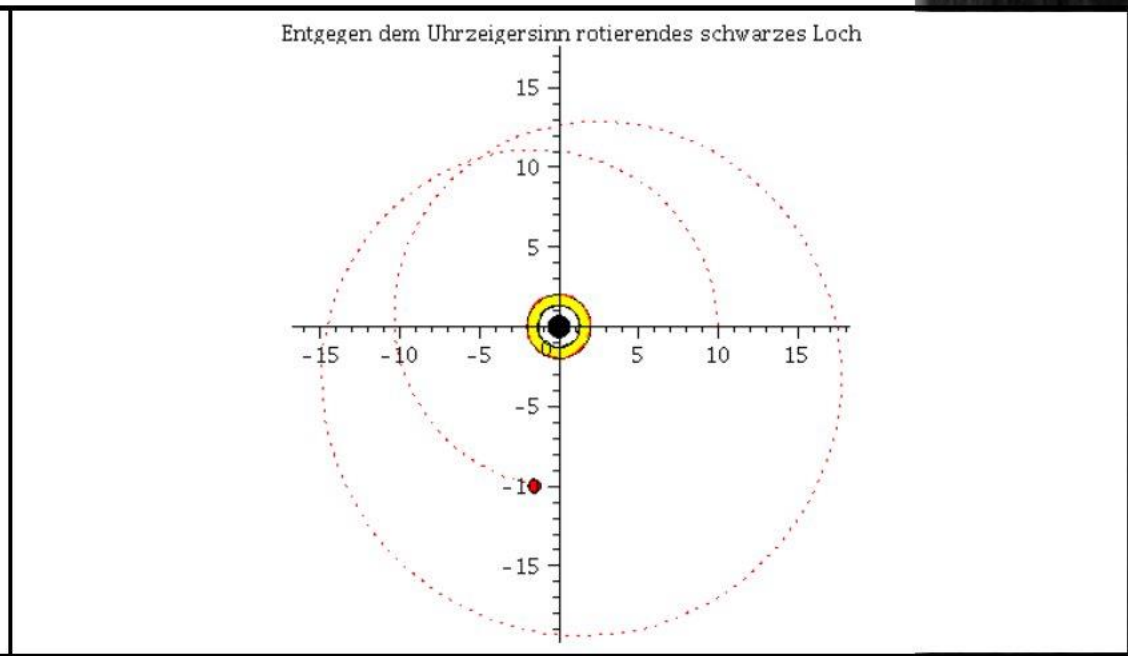
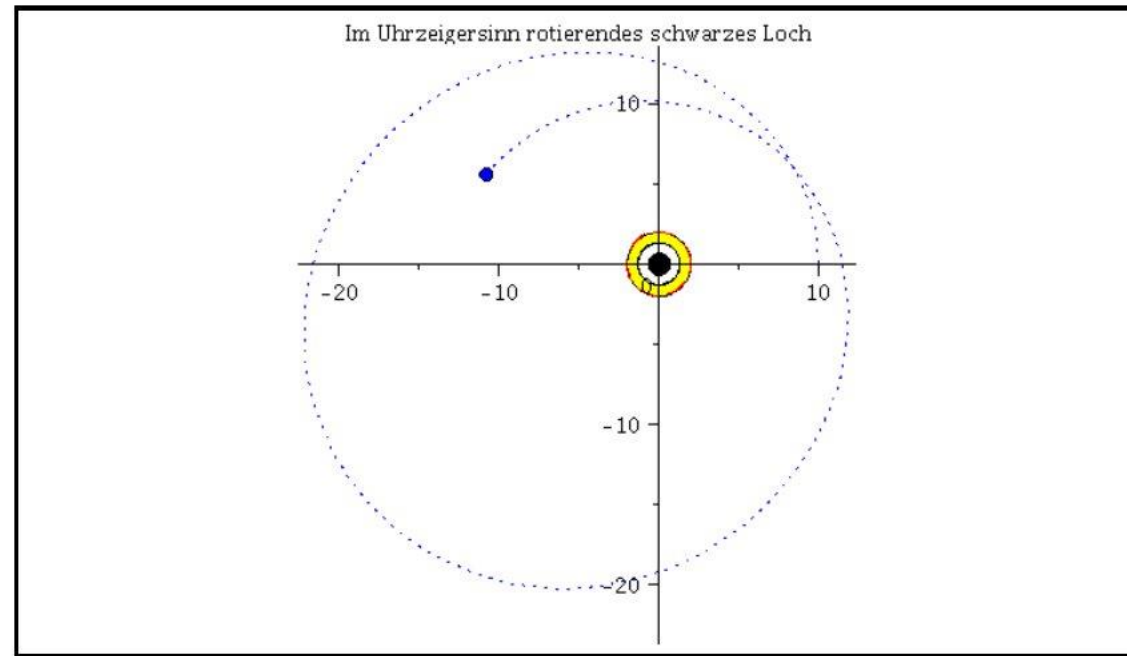
Numerisches Lösen der Geodätengleichung:



```
> display([seq(Ani[i],i=0..frames)],insequence=true);
```

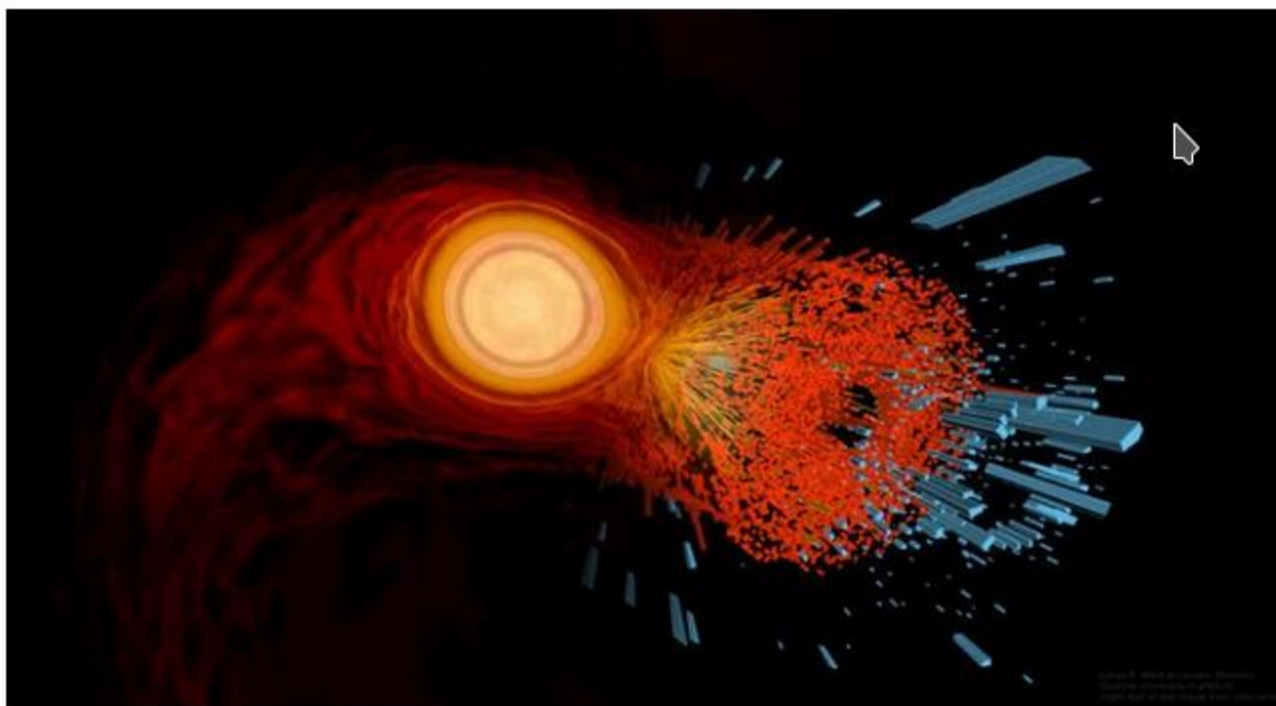


```
> Animat1:=display([seq(Ani[i],i=0..frames)],insequence=true,scaling=constrained,title="Im Uhrzeigersinn rotierendes schwarzes Loch");  
Animat2:=display([seq(Ani1[i],i=0..frames)],insequence=true,title="Entgegen dem Uhrzeigersinn rotierendes schwarzes Loch");  
display(Array([Animat1,Animat2]));
```



## Gravitational waves could prove the existence of the quark-gluon plasma

 30. April 2020



 This is the news cha  
information from scienc  
University.

Montage of the computer simulation of two merging neutron stars that blends over with an image from heavy-ion collisions to highlight the connection of astrophysics with nuclear physics. Credit: Lukas R. Weih & Luciano Rezzolla (Goethe University Frankfurt) (right half of the image from cms.cern)



TH Cosmo Coffee

# Postmerger Gravitational-Wave Signatures of Phase Transitions in Binary Compact Star Mergers

by Matthias Hanauske

Wednesday 3 Jun 2020, 16:30 → 18:30 Europe/Zurich

Zoom only (CERN)

**Description** With the first detection of gravitational waves from a binary system of neutron stars GW170817, a new window was opened to study the properties of matter at and above nuclear-saturation density. Reaching densities a few times that of nuclear matter and temperatures up to 100 MeV, such mergers also represent potential sites for a phase transition (PT) from confined hadronic matter to deconfined quark matter. Especially during the postmerger evolution of the produced hypermassive/supramassive hybrid star (HMHS/SMHS) the occurrence of a "delayed PT" might give a clear gravitational wave signature of the production of quark matter in the present Universe, if the PT is strong enough. The appearance of a hadron to quark PT in the interior region of the HMHS/SMHS and its conjunction with the spectral properties of the emitted gravitational wave will be in the focus of this talk. The presented results are based on fully general-relativistic hydrodynamic simulations and employing several suitably constructed equation of states that include a PT (see [Phys.Rev.Lett. 122, 061101 \(2019\)](#), [Phys.Rev.Lett. 124, 171103 \(2020\)](#)).

*Zoom connection:*

<https://cern.zoom.us/j/99377496227>

Meeting ID: 993 7749 6227

Password: 636648