

# Allgemeine Relativitätstheorie mit dem Computer

*PC-POOL RAUM 01.120  
JOHANN WOLFGANG GOETHE UNIVERSITÄT  
01. JUNI, 2018*

*MATTHIAS HANAUSKE*

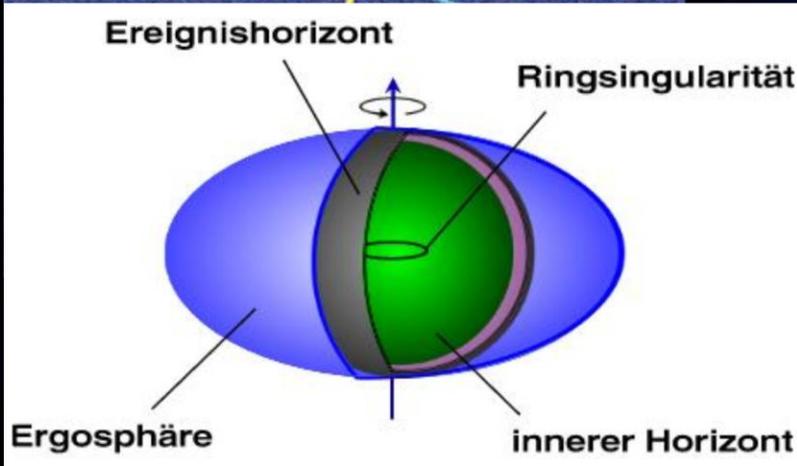
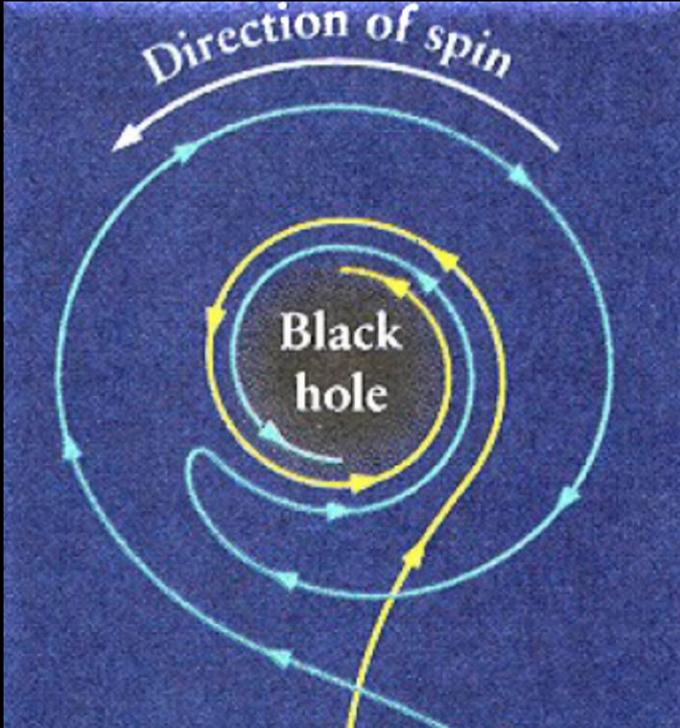
*FRANKFURT INSTITUTE FOR ADVANCED STUDIES  
JOHANN WOLFGANG GOETHE UNIVERSITÄT  
INSTITUT FÜR THEORETISCHE PHYSIK  
ARBEITSGRUPPE RELATIVISTISCHE ASTROPHYSIK  
D-60438 FRANKFURT AM MAIN  
GERMANY*

## 8. Vorlesung

# Plan für die heutige Vorlesung

- Wiederholung: Einführung in die Kerr-Metrik eines rotierenden schwarzen Loches, Ereignishorizonte und Flächen der stationären Grenze (bzw. der unendlichen Rotverschiebung), der Mitführungseffekt der Raumzeit ("Frame-Dragging"), geodätische Bewegung eines Probekörpers in der Kerr Metrik, Klassifizierung der möglichen Bahnbewegungen um ein rotierendes schwarzes Loch (Kerr Metrik) mittels eines effektiven Potentials, Kreisförmige Bewegungen in der äquatorialen Ebene
- Der "Innermost Stable Circular Orbit" für einen Probekörper der mit und entgegen der Rotationsrichtung des schwarzen Loches bewegt
- Der gravitomagnetische Effekt
- Einführung in Teil II: Numerisches Lösen der TOV-Gleichungen, Parallele Programmierung

# Rotierende schwarze Löcher und die Kerr Metrik



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

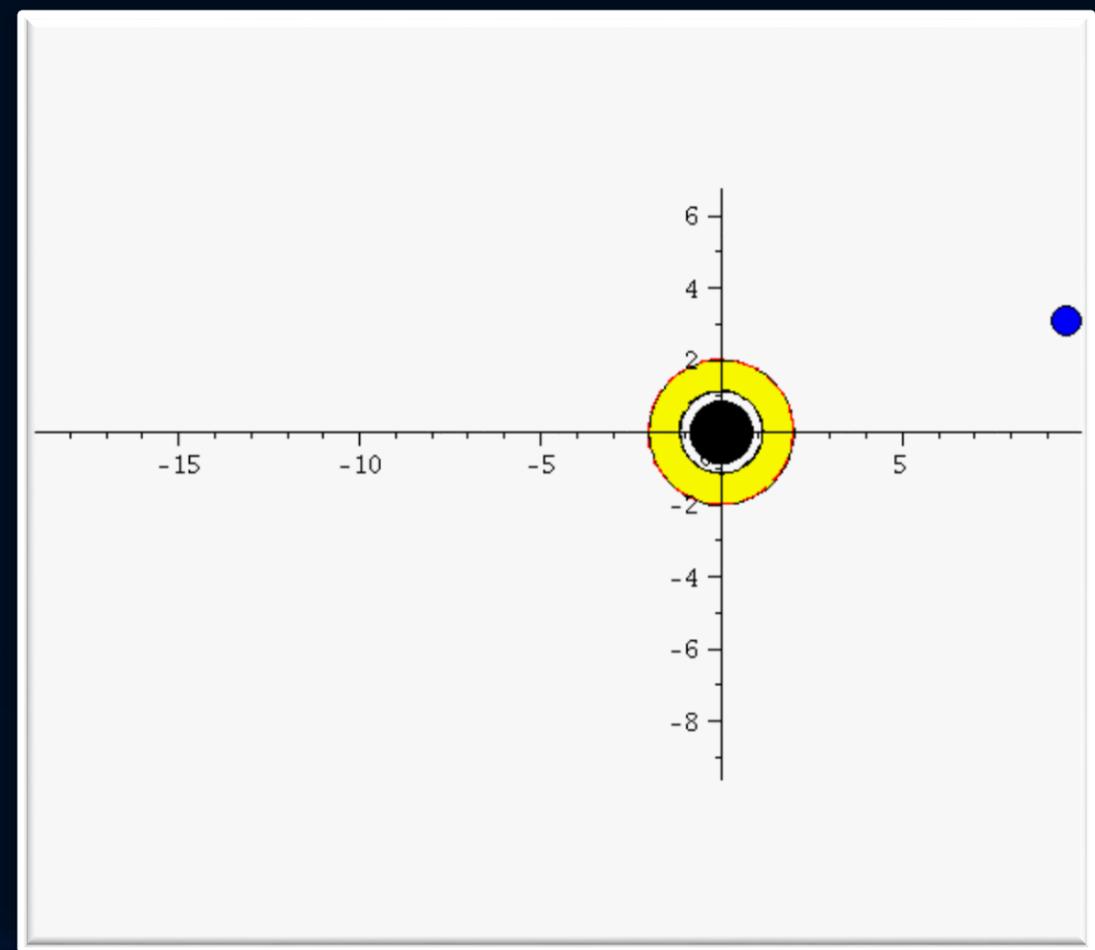
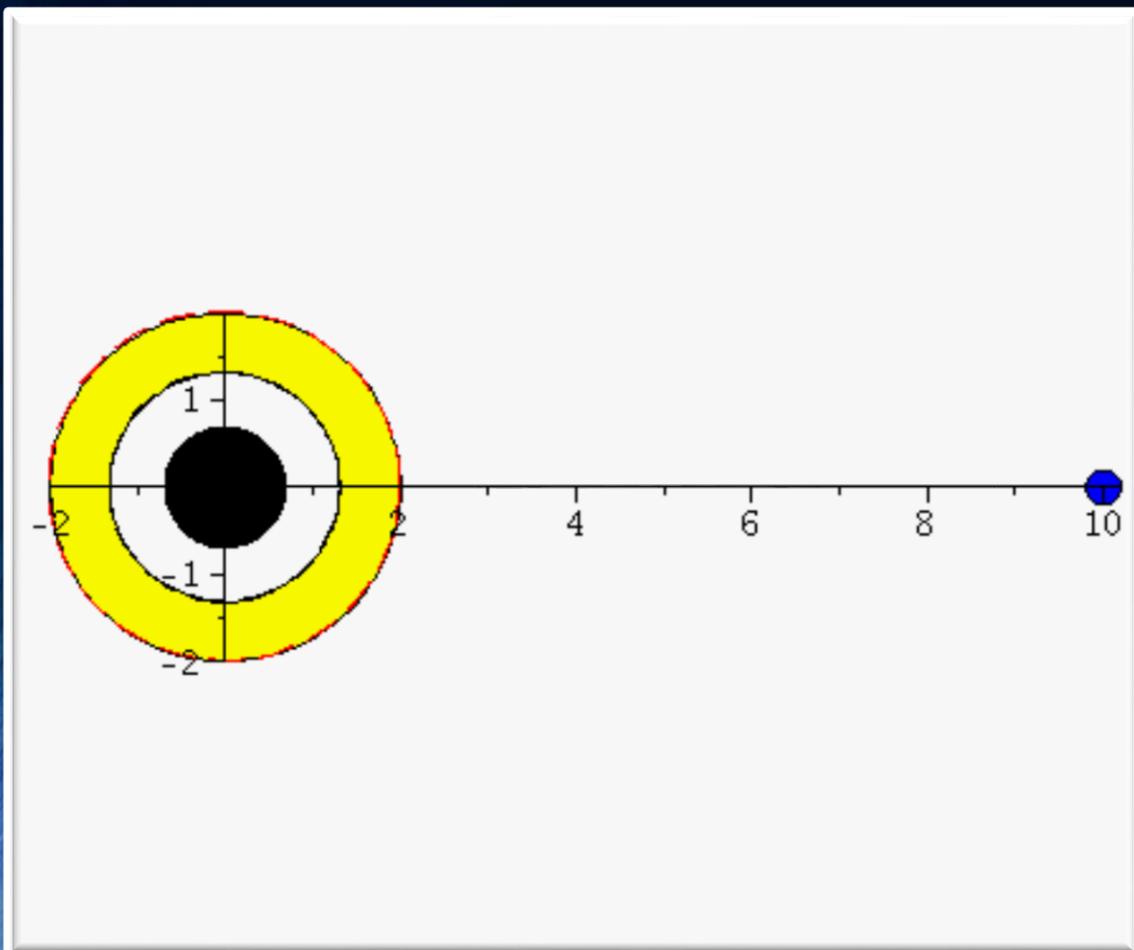
Experimente zur Bestätigung des Effektes: LARES, Gravity Probe B

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



Gravity Probe B

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

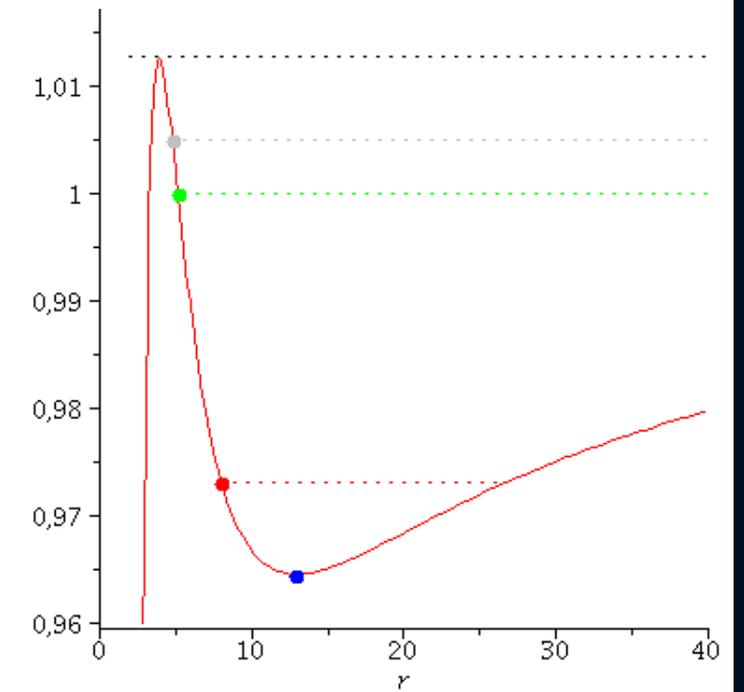
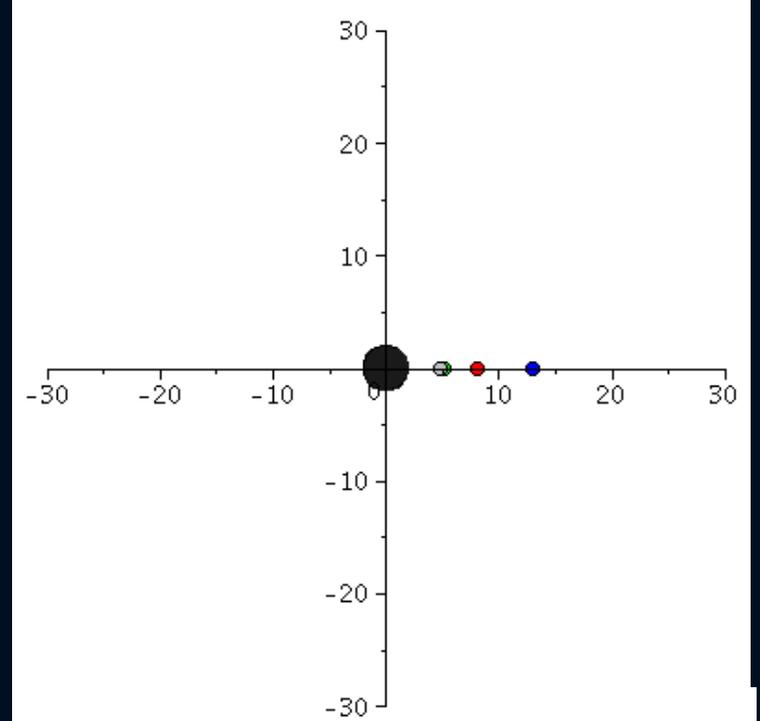
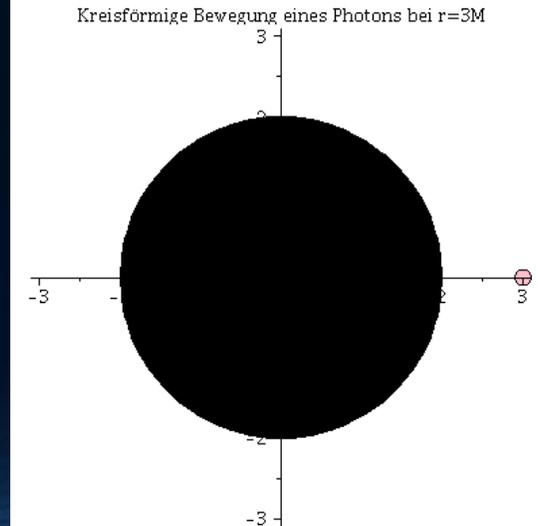
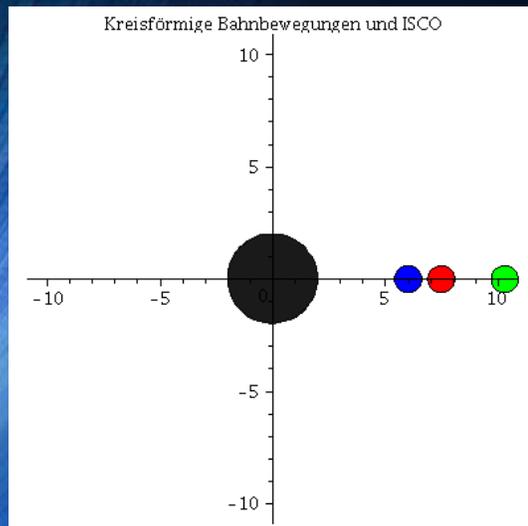


# Geodätische Bewegung um ein nicht-rotierendes schwarzes Loch

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

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

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



# Effektives Potential des nicht-rotierendes schwarzes Loch

$$4. \text{ Gleichung: } \rightarrow \frac{1}{2} \left( \frac{dr}{d\tau} \right)^2 + V(r, M, l) = \frac{1}{2} (E^2 - 1)$$

$$\text{wobei: } V(r, M, l) = \frac{l^2}{2r^2} \left( 1 - \frac{2M}{r} \right) - \frac{M}{r} ,$$

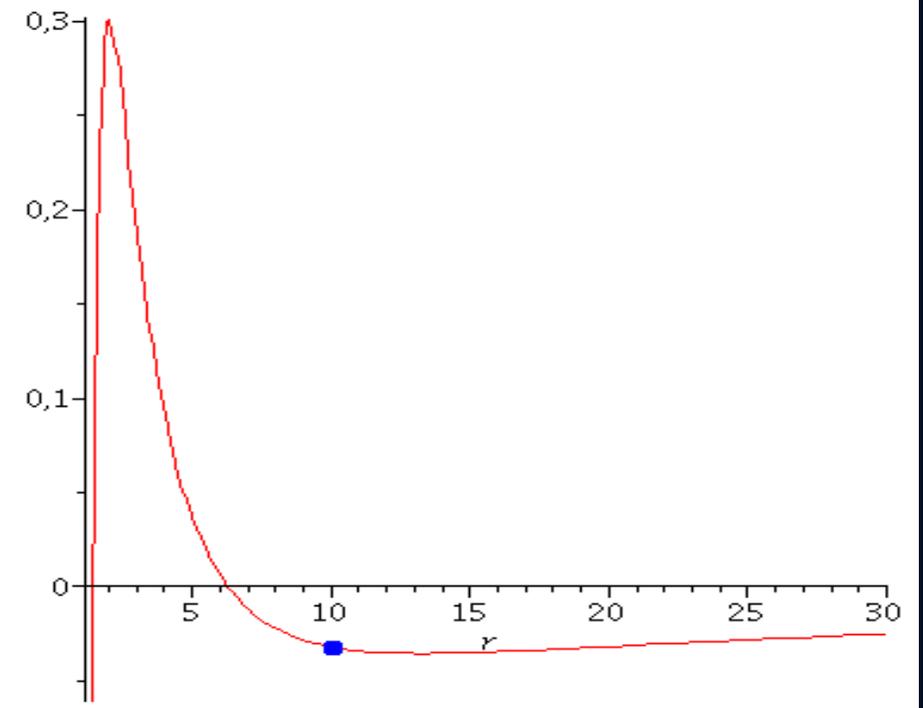
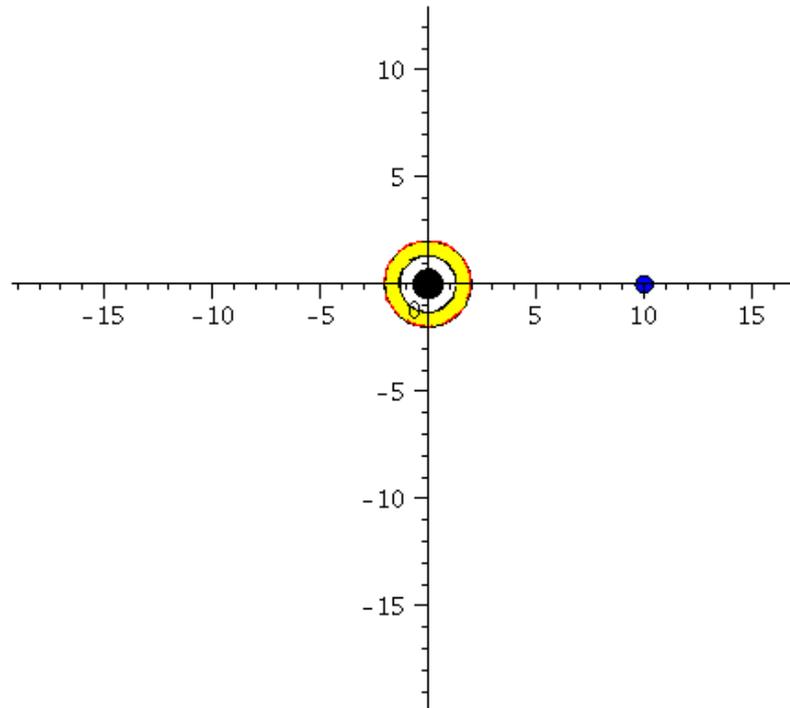
das Potential  $V(r, M, l)$  hingegen wie folgt definiert:

$$4. \text{ Gleichung: } \rightarrow \left( \frac{dr}{d\tau} \right)^2 + (V(r, M, l))^2 = E^2$$

$$\text{wobei: } V(r, M, l) = \sqrt{\left( 1 - \frac{2M}{r} \right) \left( 1 + \frac{l^2}{r^2} \right)} ,$$

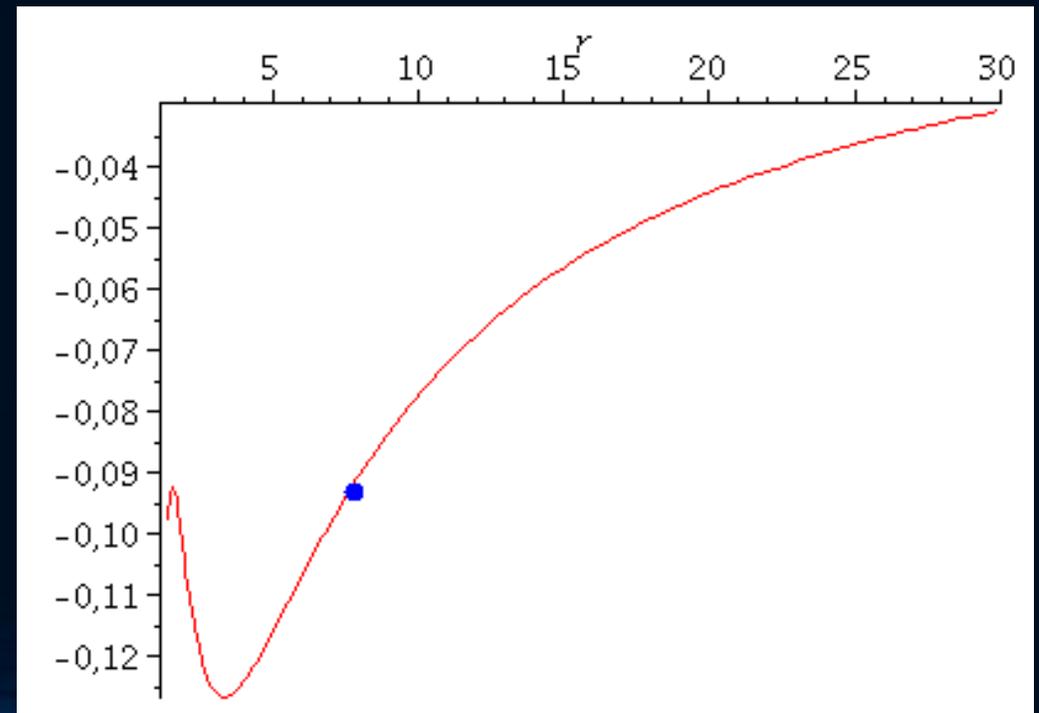
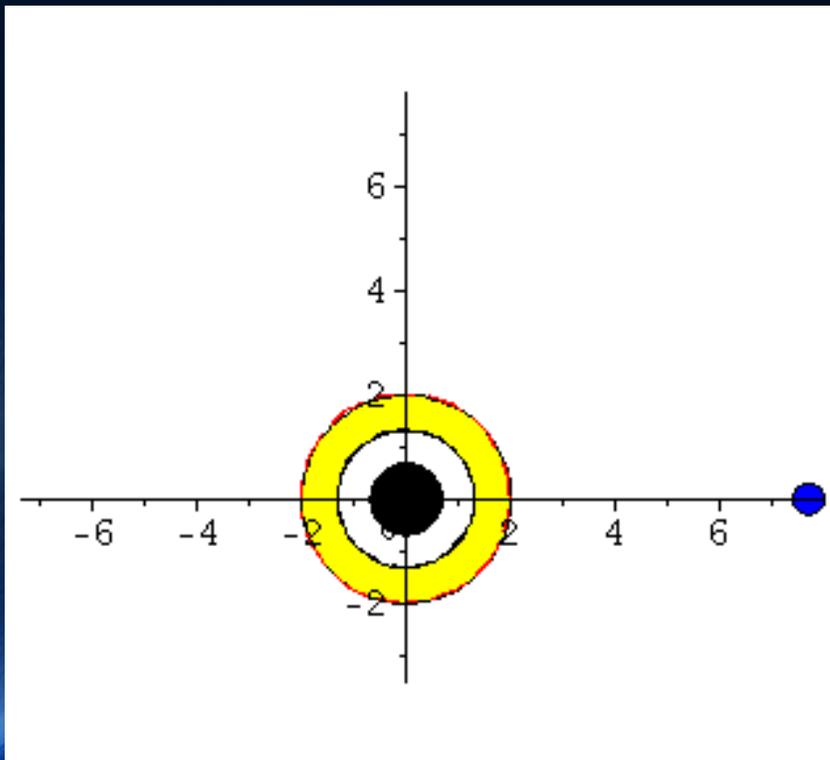
# Kerr Metrik: Effektives Potential

$$V_{eff}(r, M, l, a, E) = -\frac{M}{r} + \frac{l^2 - a^2(E^2 - 1)}{2r^2} - \frac{M(l - aE)^2}{r^3}$$



# Kerr Metrik: Effektives Potential

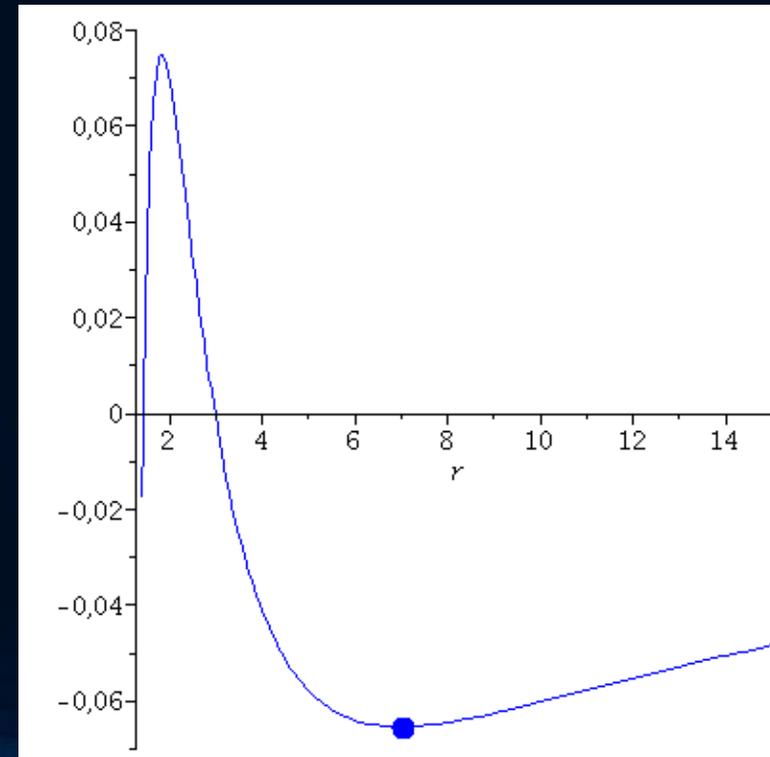
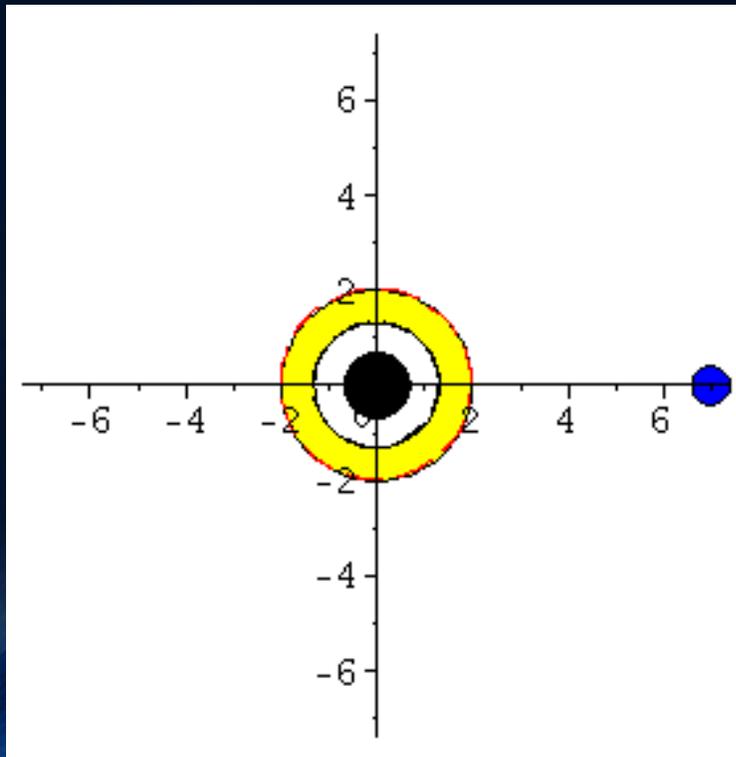
$$V_{eff}(r, M, l, a, E) = -\frac{M}{r} + \frac{l^2 - a^2(E^2 - 1)}{2r^2} - \frac{M(l - aE)^2}{r^3}$$



# Kerr Metrik: Effektives Potential

## Kreisförmige Bahnbewegungen

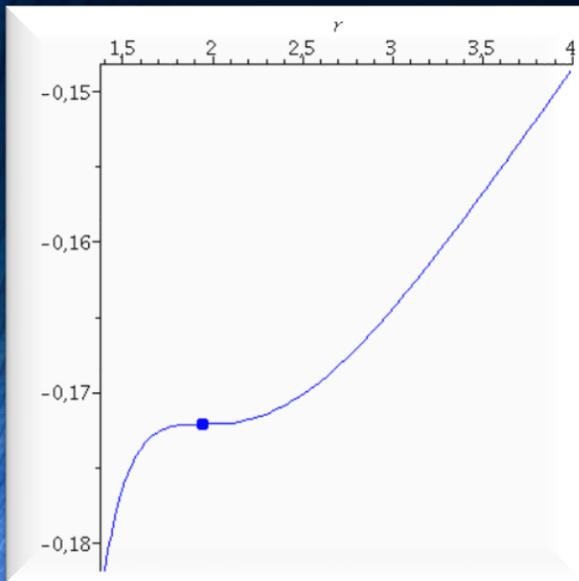
$$V_{eff}(r, M, l, a, E) = -\frac{M}{r} + \frac{l^2 - a^2(E^2 - 1)}{2r^2} - \frac{M(l - aE)^2}{r^3}$$



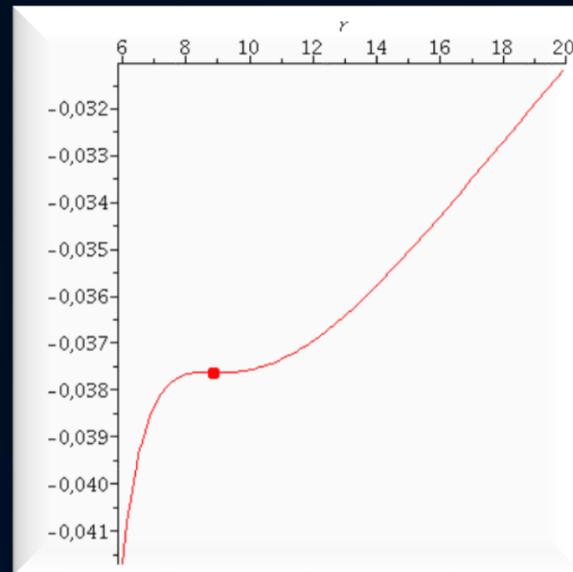
# Kerr Metrik: Effektives Potential

## Innerste „stabile“ kreisförmige Bahnbewegungen (ISCOs)

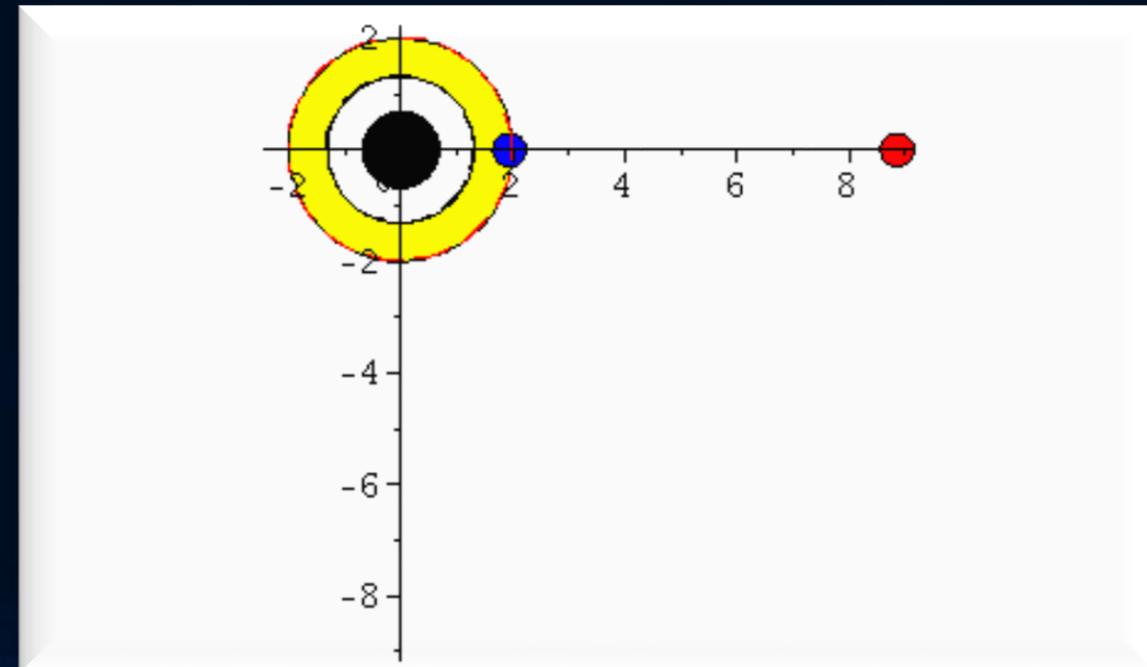
$$V_{eff}(r, M, l, a, E) = -\frac{M}{r} + \frac{l^2 - a^2(E^2 - 1)}{2r^2} - \frac{M(l - aE)^2}{r^3}$$



Probekörper rotiert mit der Rotationsrichtung des schwarzen Loches

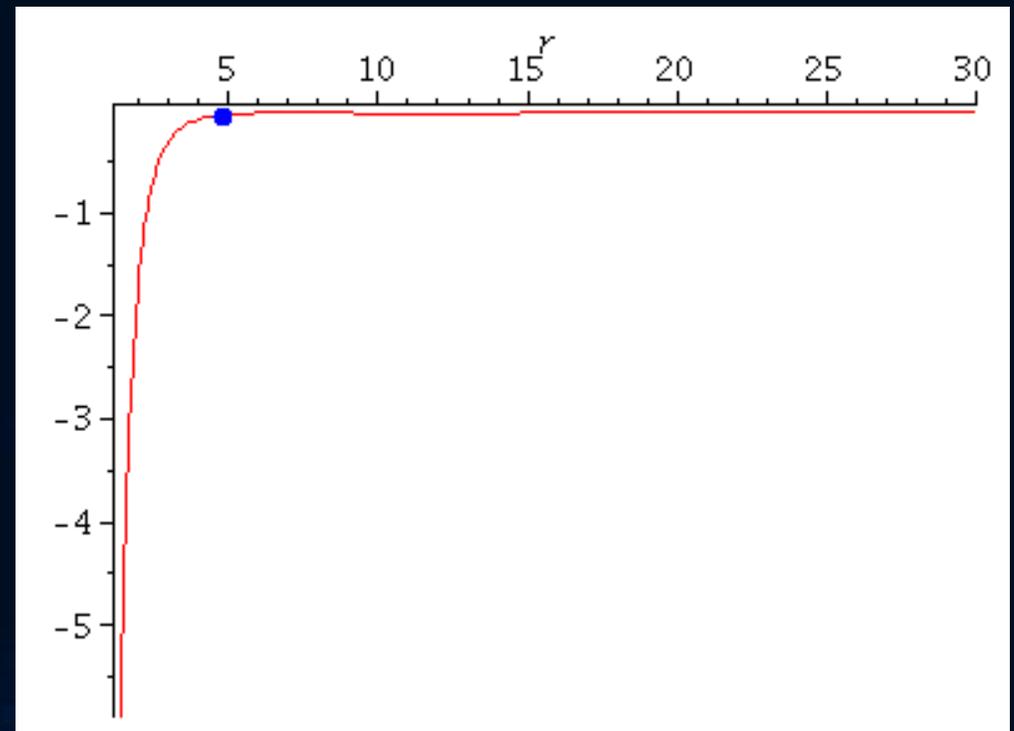
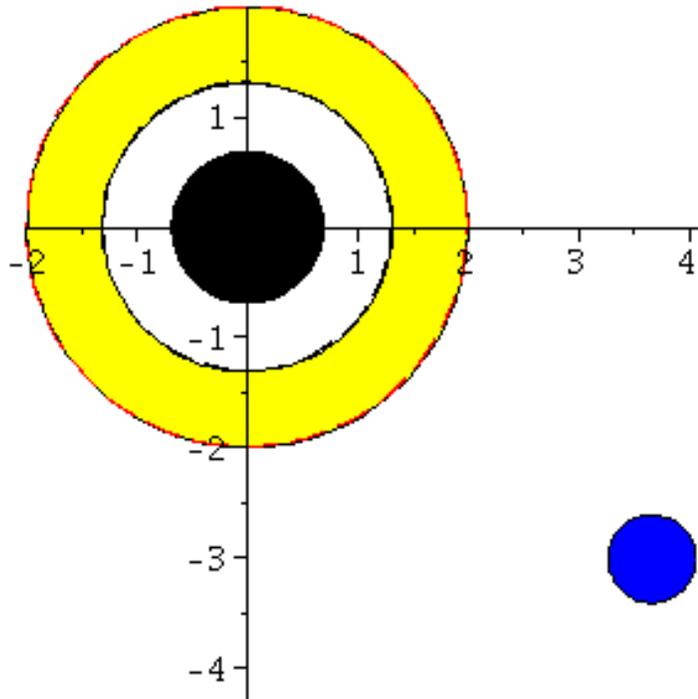


Probekörper rotiert entgegen der Rotationsrichtung des schwarzen Loches



# Kerr Metrik: Bewegungen innerhalb der Ergosphäre

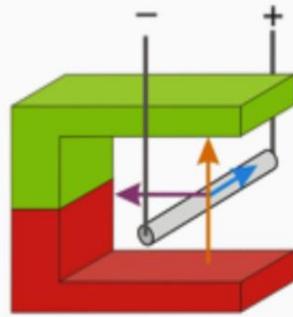
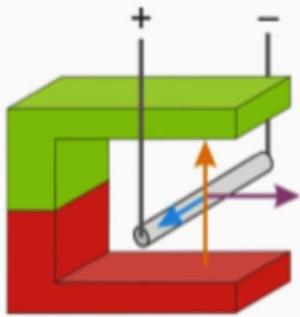
$$V_{eff}(r, M, l, a, E) = -\frac{M}{r} + \frac{l^2 - a^2(E^2 - 1)}{2r^2} - \frac{M(l - aE)^2}{r^3}$$



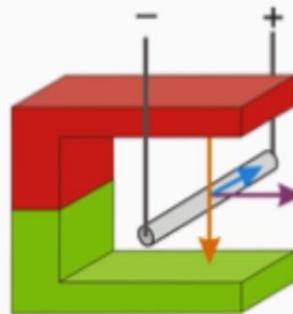
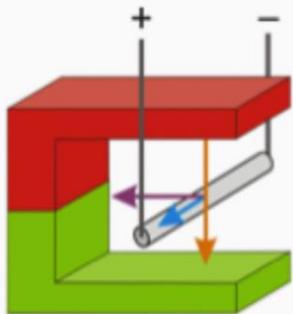
# Der gravitomagnetische Effekt

## Beobachtung

a) + b)



Versuche eine Regel mit Daumen, Zeigefinger und Mittelfinger deiner linken Hand zu formulieren.



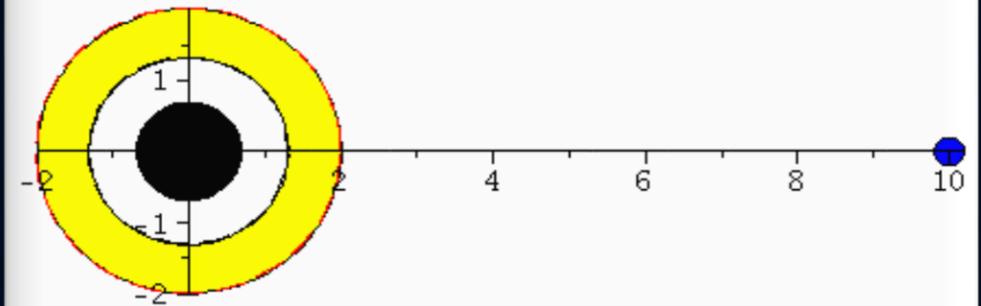
(C) Lorenz K Schröfl



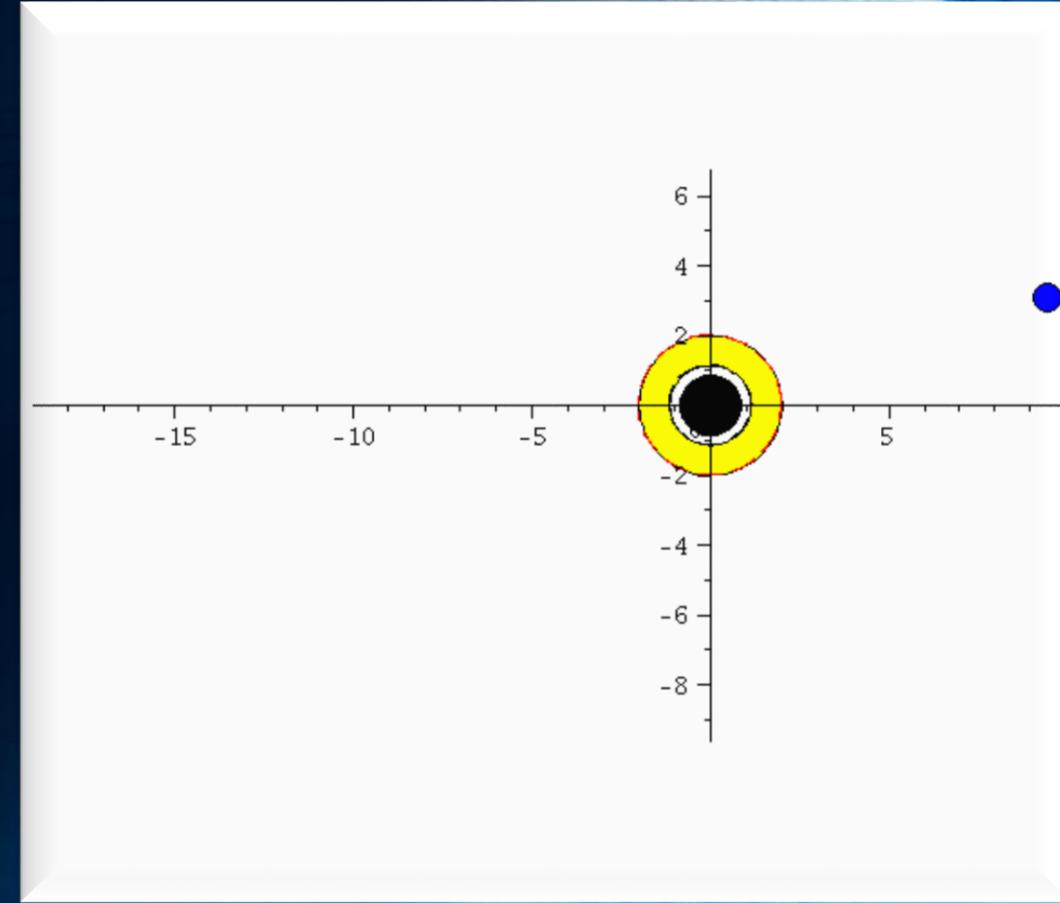
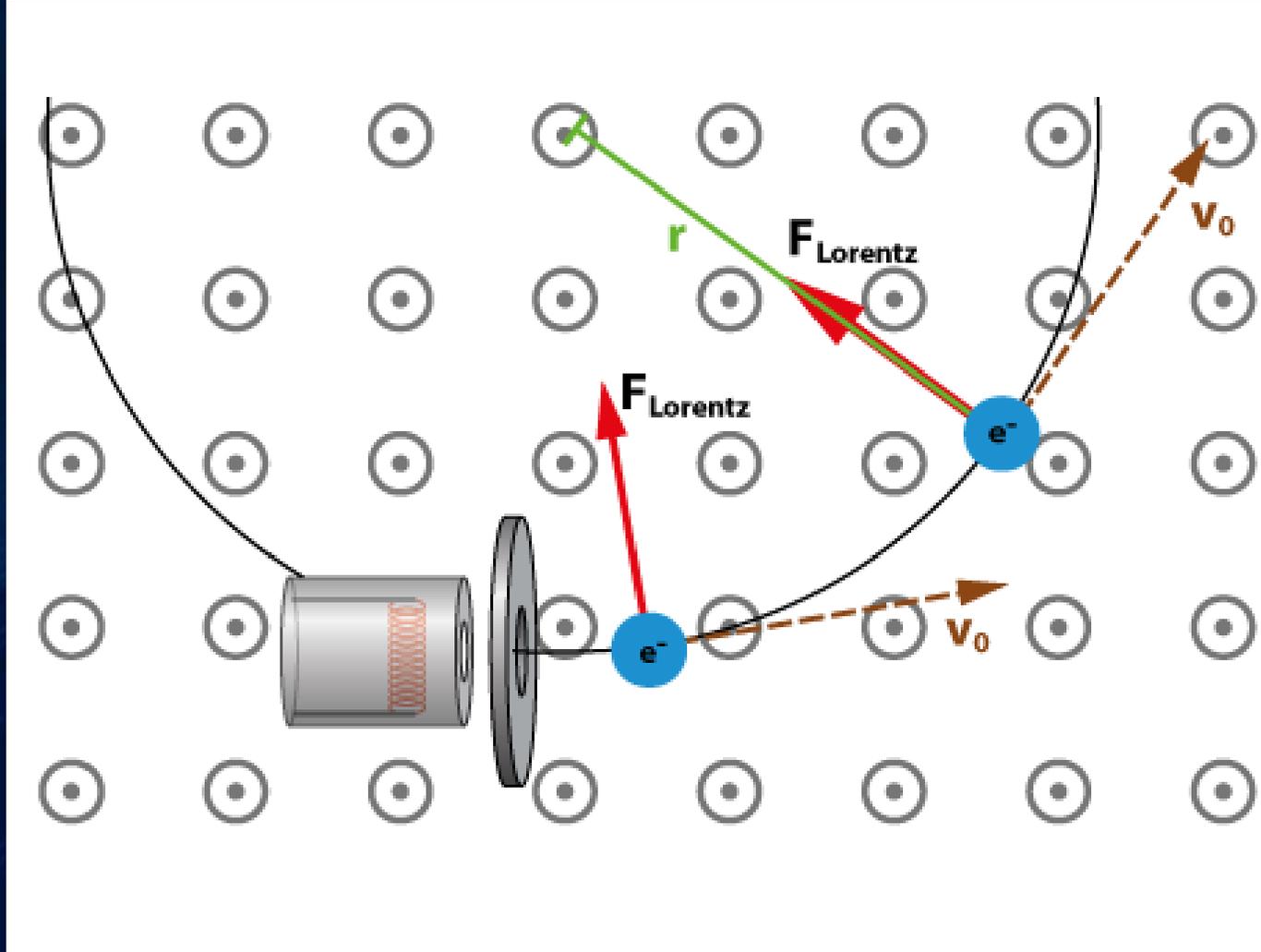
Magnetfeld geht in die Zeichenebene

Elektronenstrahl

Positronenstrahl

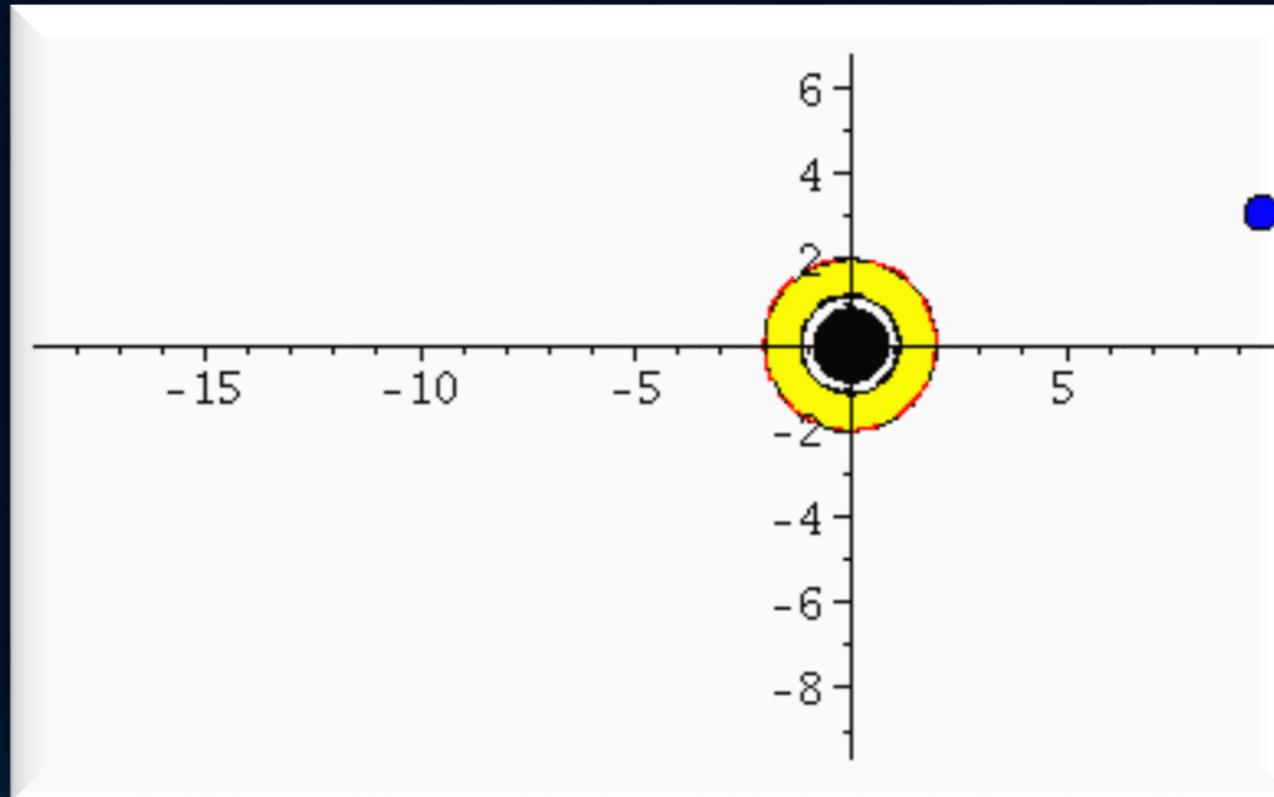


# Der gravitomagnetische Effekt



# Kerr Metrik: Der gravitomagnetische Effekt

Die grüne Kurve entspricht einer Situation ohne Magnetfeld (nur Coulombkraft = nur gravitative Anziehung, keine Rotation), die blaue Kurve entspricht einer Situation wo das gravitomagnetische Feld in +z-Richtung (schwarzes Loch rotiert entgegen dem Uhrzeigersinn) zeigt und bei der roten Kurve zeigt das gravitomagnetische Feld in -z-Richtung (schwarzes Loch rotiert im Uhrzeigersinn).





[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 beschrieben 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.

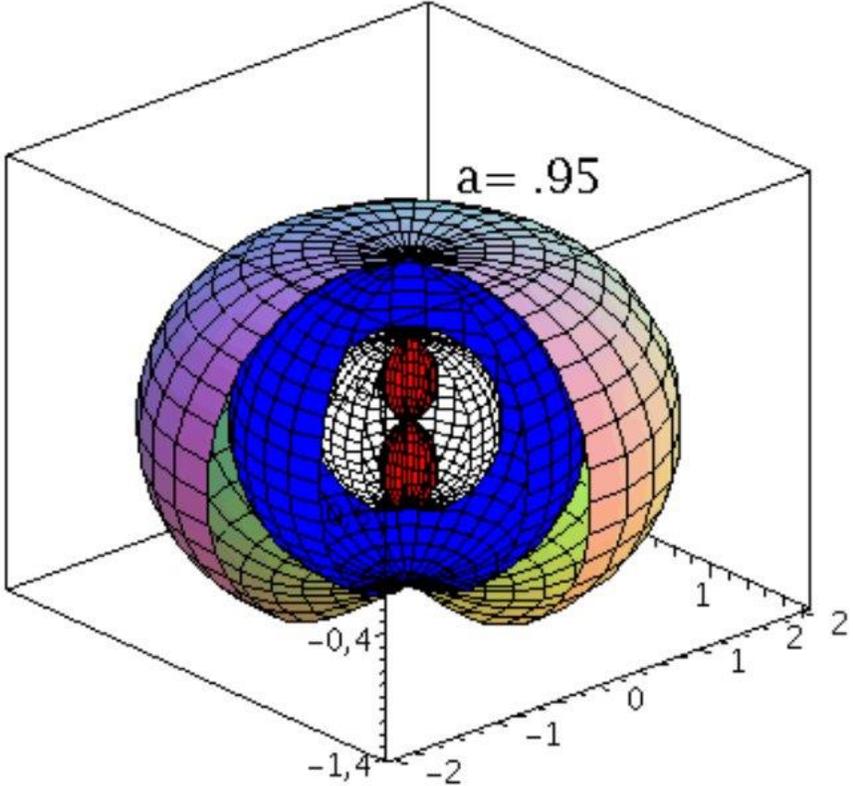
```
> UnRot:=solve(get_compts(g)[1,1]=0,r);
```

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

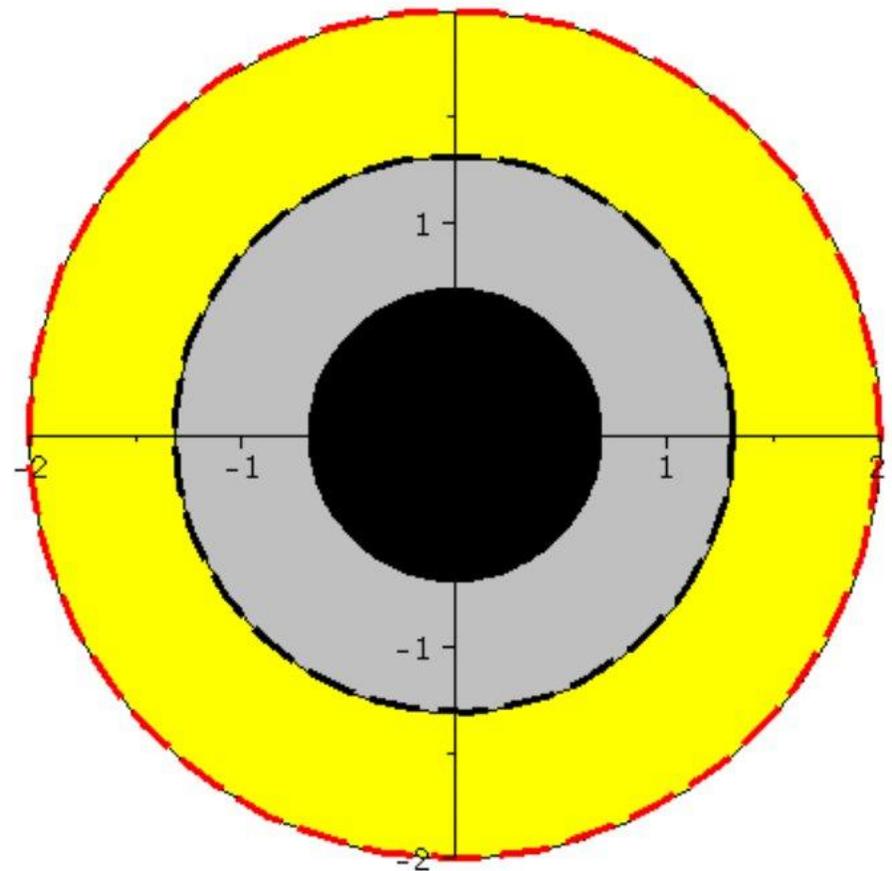
Die Ereignishorizonte sind durch  $g^{rr} = 0$  (bzw.  $g_{rr} \rightarrow \infty$ ) bestimmt. Man erhält wieder zwei Lösungen, die man gewöhnlich mit den Symbolen  $r_+$  und  $r_-$  bezeichnet.

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

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

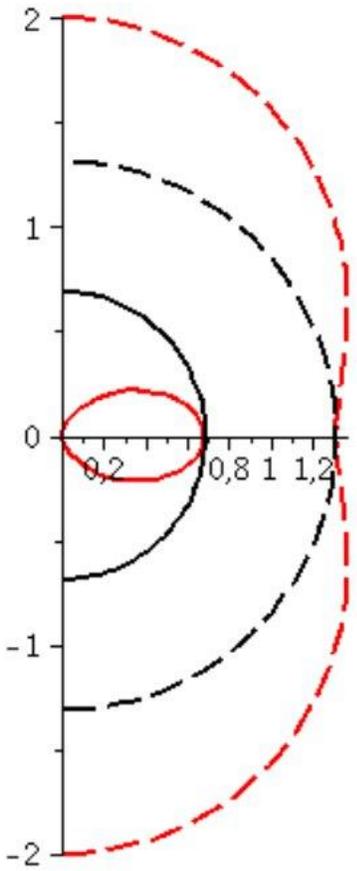


Horizontstruktur in der äquatorialen Ebene (Gelb: Ergosphäre, grau: Bereich zwischen äußerem und innerem Ereignishorizont,  $a=0.95$ ):



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
---	----	---	----	---	----

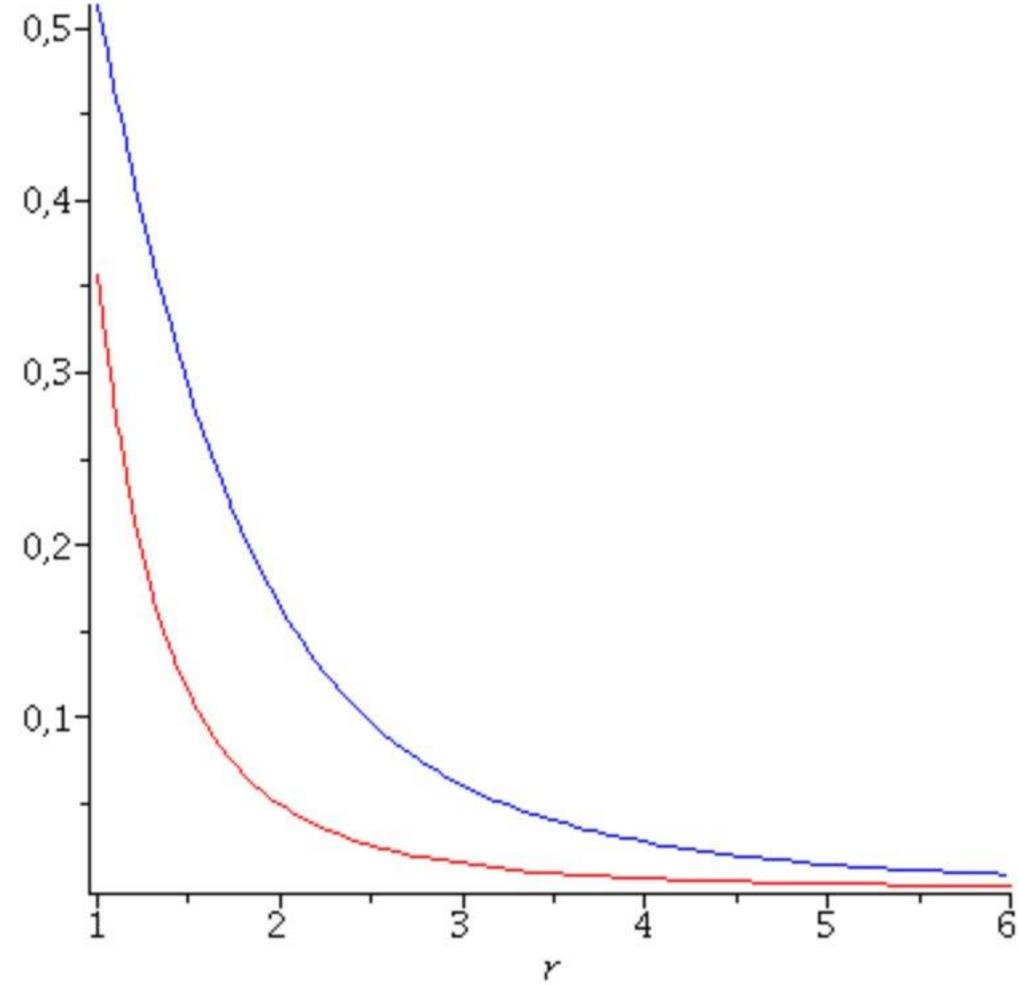
## ***Die Rotation der raumzeitlichen Struktur um das schwarze Loch (das Frame dragging)***

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:

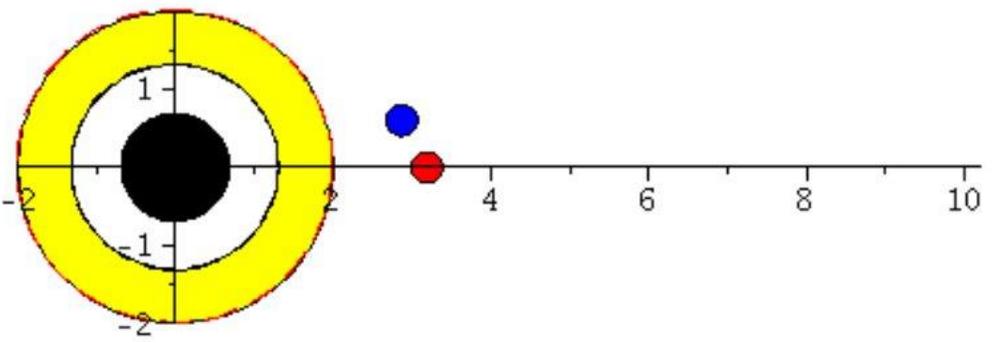
```
> dt=solve(evalf(subs({M=setM,a=seta,theta=Pi/2,dr=0,dtheta=0,dphi=0,r=10},ds2=1)),dt);
```

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

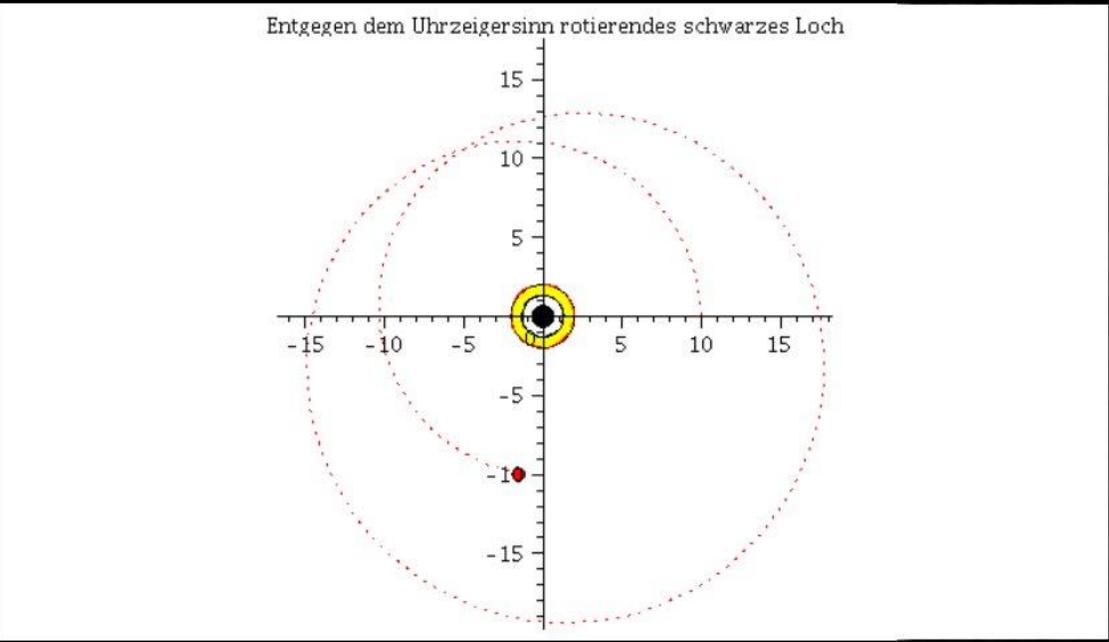
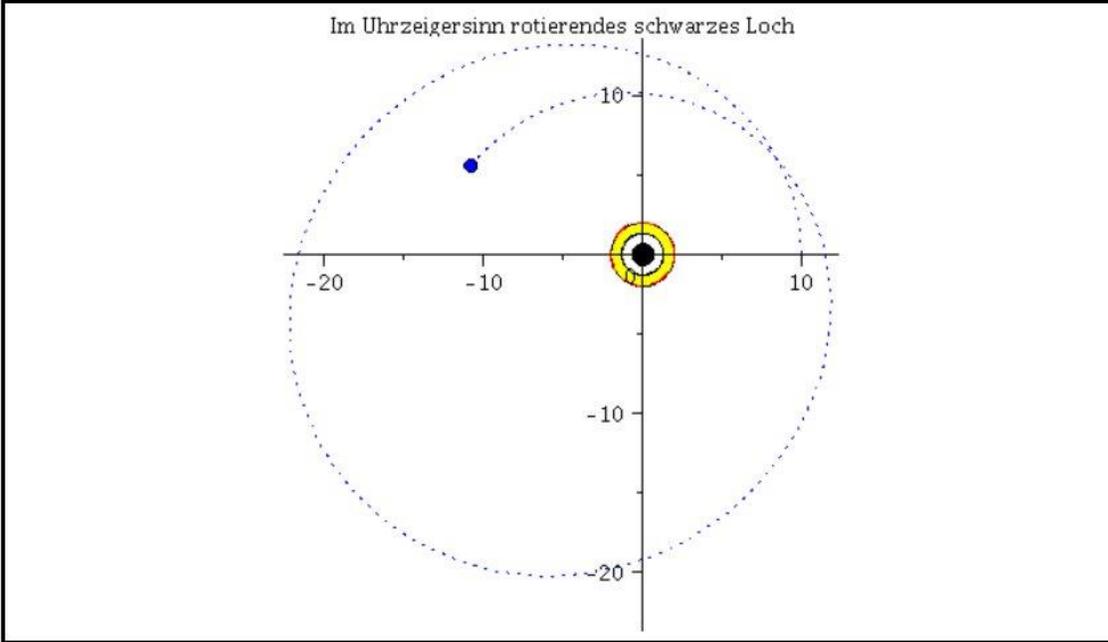
```
>      r0:=10:
      t0:=0:
      theta0:=Pi/2:
      phi0:=0:
      dr0:=0:
      dt0:=evalf(1/sqrt(1-2/r0)):
      dtheta0:=0:
      dphi0:=0:
```

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]));
```



# 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

6. Vorlesung

### **Die Kerr Metrik: Effektives Potential, kreisförmige Bewegungen, die innerste stabile Kreisbahn und der gravitomagnetische Effekt**

Basierend auf den Ergebnissen der geodätischen Bewegung eines Probekörpers um ein rotierendes Kerr schwarzes Loch (siehe Vorlesung 5), werden die möglichen Bewegungen mittels eines definierten effektiven Potential näher verstanden. Zusätzlich wird der durch den Mitführungseffekt der Raumzeit ("Frame-Dragging" bzw. Lense-Thirring Effekt) verursachte, gravitomagnetische Effekt an einem speziellen Beispiel veranschaulicht.

#### **Bahnbewegungen in der Ebene und das effektive Potential $V(r)$**

ähnlich wie im nichtrotierenden Fall (siehe Vorlesung 3) charakterisieren wir die unterschiedlichen Bahnbewegungen mittels eines effektiven Potentials:

Berechnung der Geodätengleichung als Funktion des affinen Parameters  $\lambda$ : Die Geodätengleichung ist ein System gekoppelter Differentialgleichungen

$$\begin{aligned} \frac{d^2 t}{d\lambda^2} &= -\Gamma_{\nu\rho}^0 \frac{dx^\nu}{d\lambda} \frac{dx^\rho}{d\lambda} \\ \frac{d^2 r}{d\lambda^2} &= -\Gamma_{\nu\rho}^1 \frac{dx^\nu}{d\lambda} \frac{dx^\rho}{d\lambda} \\ \frac{d^2 \theta}{d\lambda^2} &= -\Gamma_{\nu\rho}^2 \frac{dx^\nu}{d\lambda} \frac{dx^\rho}{d\lambda} \\ \frac{d^2 \phi}{d\lambda^2} &= -\Gamma_{\nu\rho}^3 \frac{dx^\nu}{d\lambda} \frac{dx^\rho}{d\lambda} \end{aligned} ,$$

wobei  $\lambda$  ein affiner Parameter (z.B. die Eigenzeit),  $t$ ,  $r$ ,  $\theta$  und  $\phi$  die sphärischen Koordinaten und  $\Gamma_{\nu\rho}^\mu$  die Christoffel Symbole zweiter Art darstellen.

> eqns:=geodesic\_eqns( coord, lambda, Cf2 );

$$\begin{aligned} eqns := & \left[ \frac{d^2}{d\lambda^2} \phi(\lambda) - \left( 2 \sin(\theta)^2 M a (-r^2 + a^2 \cos(\theta)^2) \left( \frac{d}{d\lambda} t(\lambda) \right) \left( \frac{d}{d\lambda} r(\lambda) \right) \right) \right] / \\ & (a^6 \cos(\theta)^4 + a^4 \cos(\theta)^4 r^2 + 2 r^2 a^4 \cos(\theta)^2 + 2 r^4 a^2 \cos(\theta)^2 \\ & - 4 M r^3 a^2 \cos(\theta)^2 + r^6 + r^4 a^2 - 2 M r^5 - 4 M^2 r^2 a^2 \cos(\theta)^2 - 2 \cos(\theta)^4 M r a^4) \end{aligned}$$

```

eq1:=simplify(subs({r(lambda)(lambda)=r(lambda),theta(lambda)(lambda)=theta(lambda)},eq1)):
eq2:=simplify(subs({r(lambda)(lambda)=r(lambda),theta(lambda)(lambda)=theta(lambda)},eq2)):
eq3:=simplify(subs({r(lambda)(lambda)=r(lambda),theta(lambda)(lambda)=theta(lambda)},eq3)):
eq4:=simplify(subs({r(lambda)(lambda)=r(lambda),theta(lambda)(lambda)=theta(lambda)},eq4)):

```

Infinitesimales Weglängenelement  $ds^2$ :

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu$$

```

> dx:=create([1], array([dt,dr,dtheta,dphi]));
ds2:=get_compts(prod(dx,lower(g,dx,1),[1,1]));
ds2:=collect(simplify(ds2), [dt,dr,dtheta,dphi]):
ds2a:=simplify(coeff(ds2, dt, 2)):
ds2b:=simplify(coeff(ds2, dr, 2)):
ds2c:=simplify(coeff(ds2, dtheta, 2)):
ds2d:=simplify(coeff(ds2, dphi, 2)):
ds2e:=simplify(coeff(ds2, dphi, 1)/dt):
ds2:=ds2a*dt^2+ds2e*dt*dphi+ds2b*dr^2+ds2c*dtheta^2+ds2d*dphi^2;

```

$$(-r^2 - a^2 \cos^2(\theta) + 2Mr) dt^2 - 4aMr \sin^2(\theta) dt d\phi$$

## Festlegung der Anfangswerte:

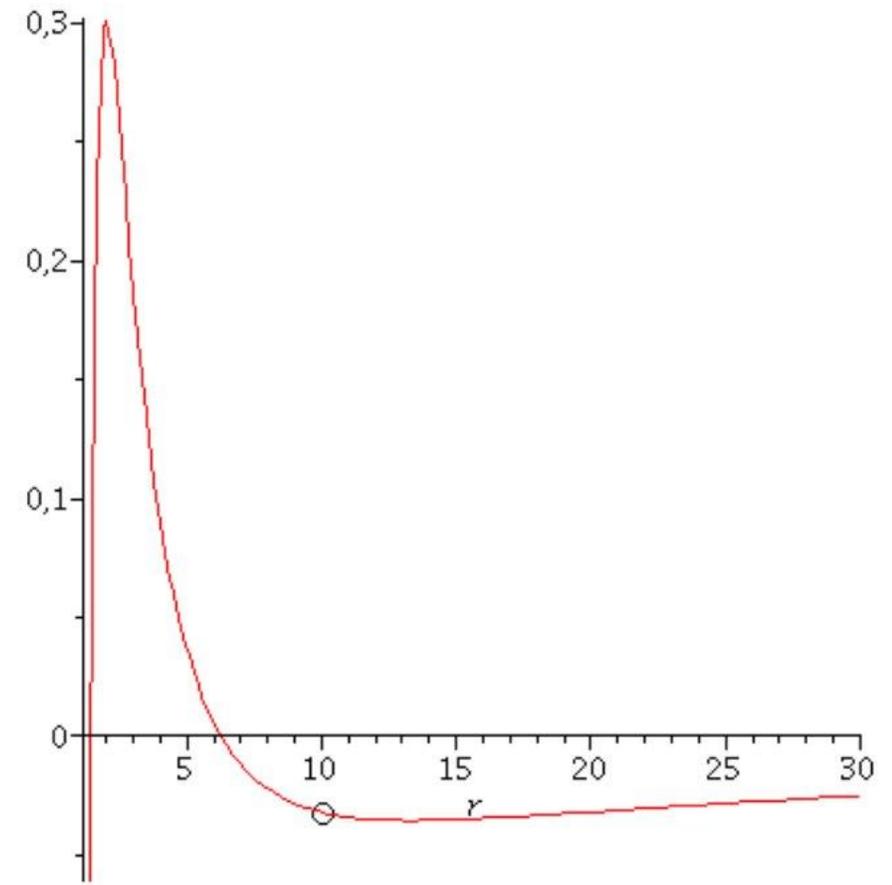
>

```
setM:=1:  
seta:=0.95:  
r0:=10:  
t0:=0:  
theta0:=Pi/2:  
phi0:=0:  
dr0:=0:  
dtheta0:=0:  
dphi0:=0.041:  
dt0:=solve(subs({theta=theta0,dphi=dphi0,dr=dr0,dtheta=dtheta0,M=1,a=0.95,r=10},ds2)=1,dt)[1]:
```

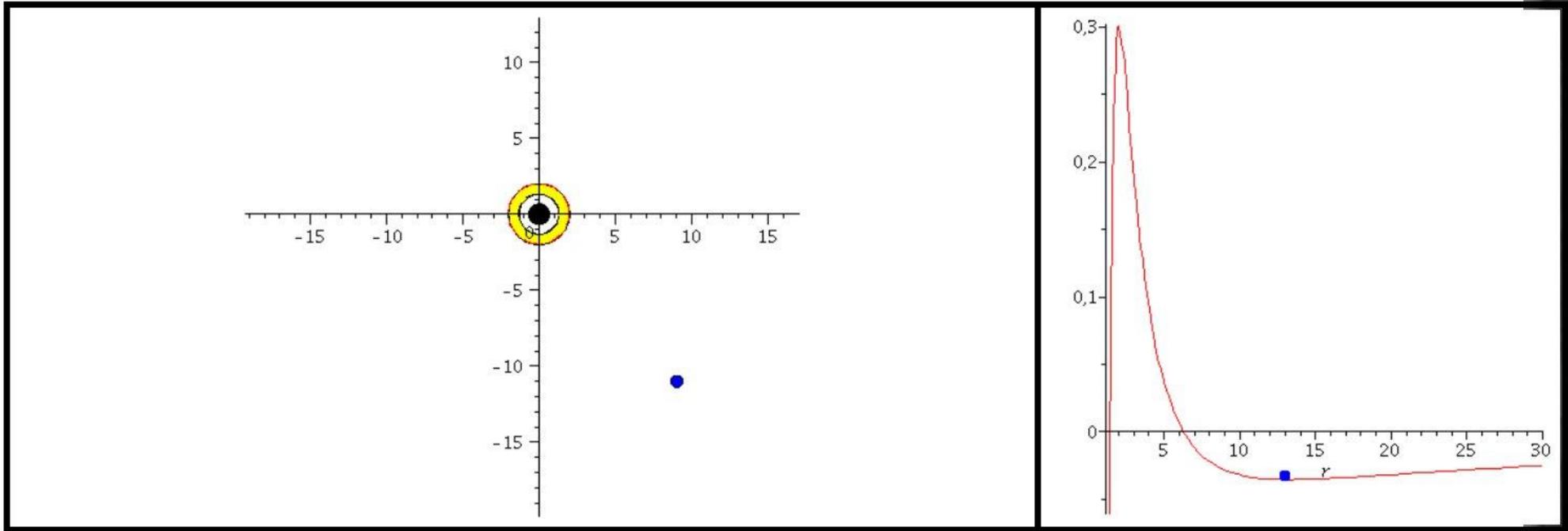
In der Literatur wird die Bewegung eines Probekörpers um ein rotierendes schwarzes Loch mittels eines definierten, effektiven Potentials illustriert (siehe z.B. Hartle- bzw. Hobson Buch). Dieses Potential hängt von dem, bei der Bewegung erhaltenem Drehimpuls pro Masse  $m$  und der Probekörper-Energie pro Masse ab. Die im Zentralfeld möglichen Bewegungen werden mittels zweier erhaltener Größen ( $l$ : Drehimpuls pro Masse  $m$  und  $E$ : Energie pro Masse) charakterisiert. Die folgende Abbildung zeigt (in der Nomenklatur vom Hartle-Buch) das effektive Potential als Funktion des Radius bei den obigen gewählten Anfangswerten:

$$V_{eff}(r, M, l, a, E) = -\frac{M}{r} + \frac{l^2 - a^2(E^2 - 1)}{2r^2} - \frac{M(l - aE)^2}{r^3}$$

$$VeffHartleRot := (r, M, l, a, en) \rightarrow -\frac{M}{r} + \frac{1}{2} \frac{l^2 - a^2 (er^2 - 1)}{r^2} - \frac{M(l - a en)^2}{r^3}$$



Numerische Lösung bei vorgegebenen Anfangswerten:



>

## Kreisförmige Bewegung eines Probekörper und der ISCO

>

## Kreisförmige Bewegung eines Probekörper und der ISCO

Wir betrachten im folgenden die kreisförmige Bewegung eines Probekörpers.

ähnlich wie im nichtrotierenden Fall (siehe Vorlesung 3) charakterisieren wir die unterschiedlichen Bahnbewegungen mittels eines effektiven Potentials:

```
> 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];
```

$$+ \frac{(-a^4 \cos(\theta)^2 - r^2 a^2 \cos(\theta)^2 - a^2 r^2 - r^4 - 2 M a^2 r + 2 M a^2 r \cos(\theta)^2) d\phi^2}{r^2 + a^2 \cos(\theta)^2}$$

Wir beschränken uns im folgenden auf äquatoriale Bewegungen. Aufgrund der Normalisierungseigenschaft der Vierergeschwindigkeit  $u^\mu u_\mu = 1$  sind die Zeit- und phi-Komponente der Viergeschwindigkeit wie folgt durch die Energie- und Drehimpulswerte bestimmt (siehe Hartle-Buch, S:318):

$$\frac{dt}{d\tau} = \frac{1}{\Delta} \left[ \left( r^2 + a^2 + \frac{2Ma^2}{r} \right) E - \frac{2Ma}{r} l \right]$$

$$\frac{d\phi}{d\tau} = \frac{1}{\Delta} \left[ \left( 1 - \frac{2M}{r} \right) l + \frac{2Ma}{r} E \right]$$

> Eqdt:=dt=1/Delta\*((r^2+a^2+2\*M\*a^2/r)\*en-2\*M\*a/r\*l);  
Eqdphi:=dphi=1/Delta\*((1-2\*M/r)\*l+2\*M\*a/r\*en);

Wir beschränken uns im Folgenden auf äquatoriale, kreisförmige Bewegungen und nehmen die folgenden Anfangsbedingungen an:

```
> t0:=0:  
  phi0:=0:  
  theta0:=Pi/2:  
  dr0:=0:  
  dtheta0:=0:
```

Bei kreisförmige Bewegungen muss die Ableitung des effektiven Potentials nach r verschwinden:

```
> VeffHartleRot:=(r,M,l,a,en)->-M/r+(l^2-a^2*(en^2-1))/(2*r^2)-M*(l-a*en)^2/r^3;  
  EqCirc:=diff(VeffHartleRot(r,M,l,a,en),r)=0;
```

$$Eqds2 := -\frac{(-r^2 + 2Mr) dt^2}{r^2} + \frac{4aM dt d\phi}{r} + \frac{(-2Mdr - r^4 - a^2 r^2) d\phi^2}{r^2} = 1 \quad (1.2.5)$$

Einsetzen der obigen Gleichungen für dt und dphi ergibt:

> `EqA:=simplify(subs({Eqdt,Eqdphi},Eqds2));`

$$EqA := -\frac{2Ml^2 + 2Mer^2 a^2 - 4Menal + er^2 r^3 - r^2 + er^2 a^2 r}{r(-r^2 + 2Mr - a^2)} = 1 \quad (1.2.6)$$

Wir nehmen nun einen Spezialfall an und setzen  $d\phi_0=0.066$  für ein im Uhrzeigersinn rotierenden Probekörper und verarbeiten dies in die vorige Gleichung:

> `dphi0:=0.066;`  
`EqB:=simplify(subs(l=solve(subs(dphi=dphi0,Eqdphi),l),EqA));`

## ***Die innerste stabile kreisförmige Bahnbewegung (Innermost Stable Circular Orbit (ISCO)) eines Probekörpers***

>

Wir wollen nun, ähnlich wie beim nichtrotierenden Schwarzschild schwarzen Loch, die innerste stabile kreisförmige Bahn (ISCO) berechnen. Das effektive Potential ist bei kreisförmigen Bahnbewegungen ( $dr/d\lambda=0$ ) wie folgt mit der Energie verknüpft :

>  $V_{\text{effHartleRot}}(r, M, l, a, e) = (e^2 - 1)/2;$

$$-\frac{M}{r} + \frac{1}{2} \frac{l^2 - a^2 (e^2 - 1)}{r^2} - \frac{M(l - ae)^2}{r^3} = \frac{1}{2} e^2 - \frac{1}{2} \quad (1.2.1.1)$$

Kreisförmige Bahnbewegungen sind dort, wo das effektive Potential sein Minimum hat:

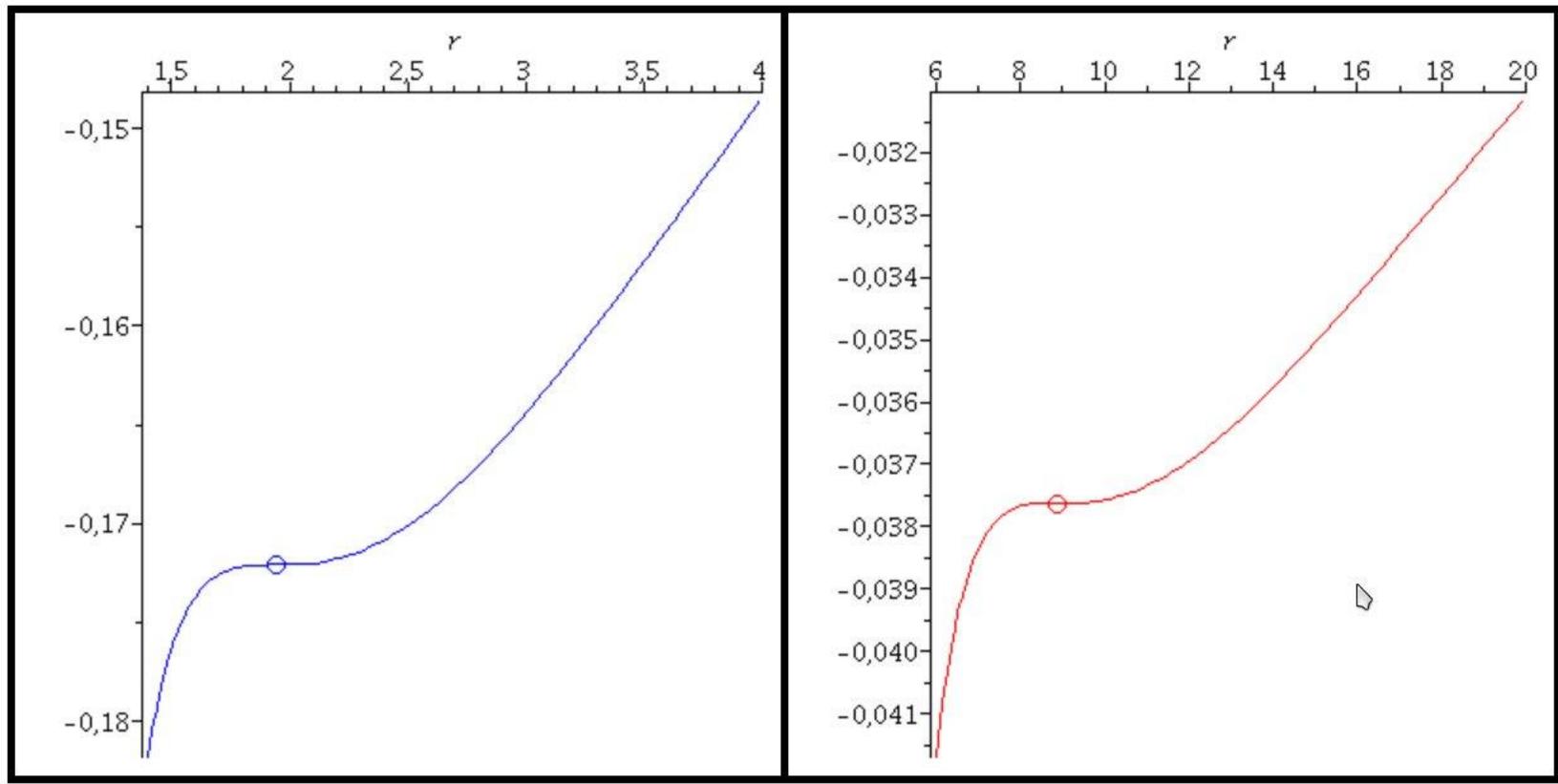
Die innerste stabile kreisförmige Bahn hat dort zusätzlich noch einen Sattelpunkt:

```
> diff(VeffHartleRot(r,M,l,a,e),r,r)=0;
```

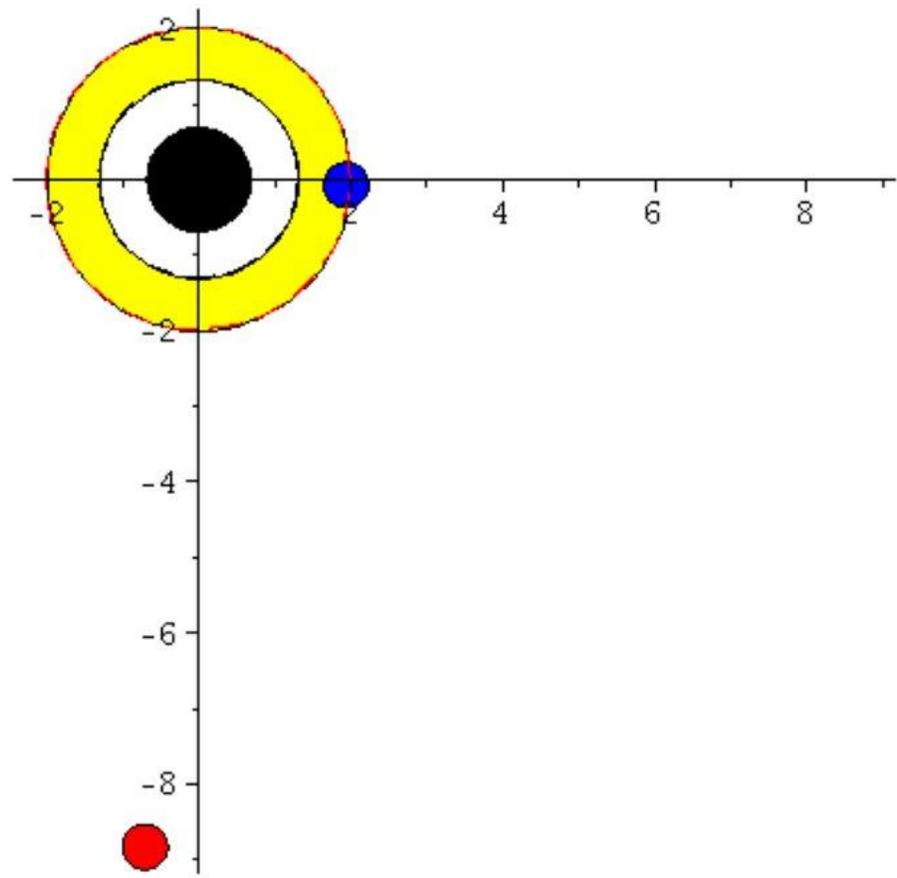
$$-\frac{2M}{r^3} + \frac{3(l^2 - a^2(e^2 - 1))}{r^4} - \frac{12M(l - ae)^2}{r^5} = 0 \quad (1.2.1.3)$$

Setzt man die Eigenschaften des rotierenden schwarzen Loches (z.B.  $M=1$  und  $a=0.95$ ) fest, und löst drei Gleichungen, so erhält man die Werte für den ISCO.

```
> setM:=1;
   seta:=0.95;
   ISCO:=solve({diff(VeffHartleRot(r,setM,l,seta,e),r)=0,diff(VeffHartleRot(r,setM,l,seta,e),r,r)=0,VeffHartleRot
   (e^2-1)/2});
```



Setzen der Anfangswerte beider Lösungen:



Man erkennt, dass der ISCO nur eine metastabile Lösung ist, da es sich um einen Sattelpunkt des effektiven Potentials handelt (blauer Probekörper fällt nach ca.  $\lambda=36.73$  in das schwarze Loch). Die blaue Kurve (Probekörper rotiert in gleicher Richtung wie die Rotationsrichtung des schwarzen Loches) hält sich zusätzlich noch innerhalb der Ergosphäre auf ( $r=1.94$  km).

## Der gravitomagnetische Effekt

Die durch die Rotation des schwarzen Lochs hervorgerufene Erscheinungen sind den magnetischen Eigenschaften in der Elektrodynamik sehr ähnlich. Der Mitführungseffekt der Raumzeit ("Frame-Dragging" bzw. Lense-Thirring Effekt) verursacht, gravitomagnetische Effekte die in ähnlicher Weise wie die Lorentzkraft in der Elektrodynamik wirken. Hier verdeutlichen wir dies an einem Beispiel:

```
> 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:
```

Im folgenden setzen wir  $M=1$  und betrachten drei Fälle:

Fall 1: Schwarzes Loch rotiert nicht ( $a=0$ , grün)

Fall 2: Schwarzes Loch rotiert nahe seines extremen Wertes im Uhrzeigersinn ( $a=0.99$ , blau)

Fall 3: Schwarzes Loch rotiert nahe seines extremen Wertes entgegen dem Uhrzeigersinn ( $a=-0.99$ , rot)

>

```
setM:=1:
seta0:=0:
setaP:=0.99:
setaN:=-0.99:
t0:=0:
phi0:=Pi/10:
theta0:=Pi/2:
dtheta0:=0:
#dphi0:=0.12:
dphi0:=0.0685:
dr0:=-0.6:
r0:=10:
```

```
Eqds2:=subs({dtheta=dtheta0,dr=dr0,cos(theta)=0,sin(theta)=1},ds2)=1:
```

```
dt00:=solve(subs({theta=theta0,dphi=dphi0,dr=dr0,dtheta=dtheta0,M=setM,a=seta0,r=r0},Eqds2),dt)[1]:
```

```
dt0P:=solve(subs({theta=theta0,dphi=dphi0,dr=dr0,dtheta=dtheta0,M=setM,a=setaP,r=r0},Eqds2),dt)[1]:
```

```
dt0N:=solve(subs({theta=theta0,dphi=dphi0,dr=dr0,dtheta=dtheta0,M=setM,a=setaN,r=r0},Eqds2),dt)[1]:
```

```
EquD0:=solve(subs({theta=theta0,dphi=dphi0,dr=dr0,dtheta=dtheta0,dt=dt00,M=setM,a=seta0,r=r0},{Eqdt,Eqdphi}},{en,l}):
```

```
EquDP:=solve(subs({theta=theta0,dphi=dphi0,dr=dr0,dtheta=dtheta0,dt=dt0P,M=setM,a=setaP,r=r0},{Eqdt,Eqdphi}},{en,l}):
```

```
EquDN:=solve(subs({theta=theta0,dphi=dphi0,dr=dr0,dtheta=dtheta0,dt=dt0N,M=setM,a=setaN,r=r0},{Eqdt,Eqdphi}},{en,l}):
```

```
setE0:=rhs(EquD0[1]):
```

```
setl0:=rhs(EquD0[2]):
```

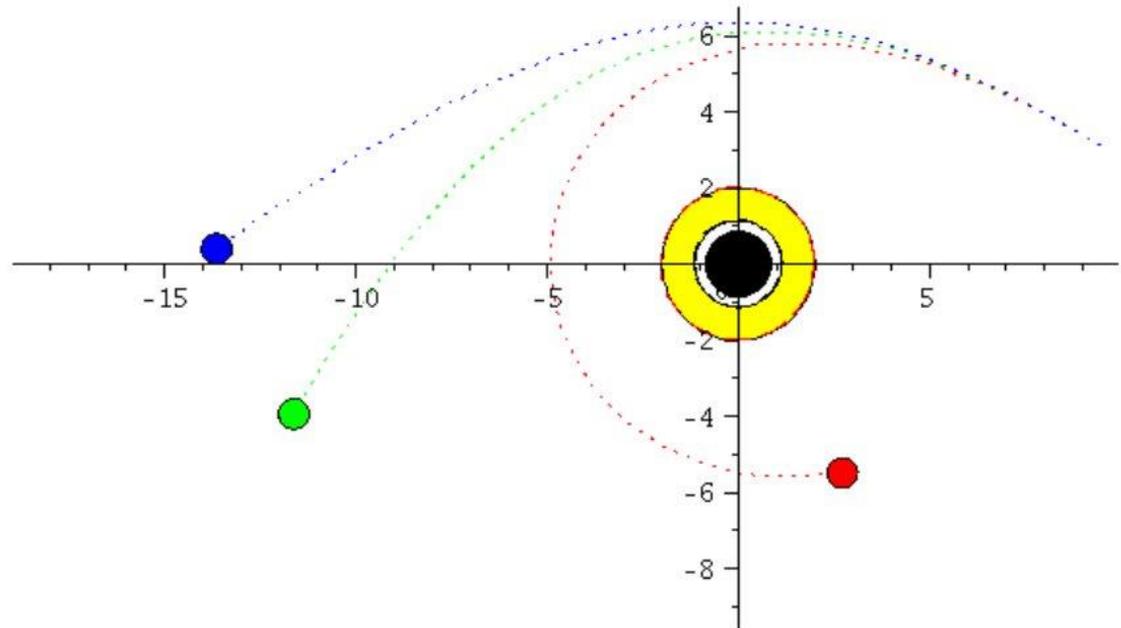
Die Rotationsfrequenz mit der die Raumzeit mitgeführt wird als "Frame dragging" Frequenz  $\omega(r, \theta)$  bezeichnet (siehe Vorlesung 5):

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

Diese "Frame dragging" Frequenz wirkt in ähnlicher Weise auf die Geschwindigkeit von Probekörpern, wie das Magnetfeld in der Elektrodynamik die Lorentzkraft verursacht. Im Fall 1 (grün) ist das gravitomagnetische Feld Null, im Fall 2 (blau) ist es aus der äquatorialen Ebene nach oben gerichtet und im Fall 3 zeigt es nach unten. In erster Näherung (siehe Fließbach Buch, S:172) ist die gravitomagnetische Lorentzkraft gleich  $\sim 2(\boldsymbol{\omega} \times \mathbf{v})$ , wobei  $\times$  das Kreuzprodukt,  $\boldsymbol{\omega}$  der axiale Vektor der "Frame dragging" Frequenz und  $\mathbf{v}$  der Geschwindigkeitsvektor des Probekörpers ist. Die Änderung des Geschwindigkeitsvektors nimmt in Schwachfeldnäherung dann die folgende Gestalt an:

$$\frac{d\mathbf{v}}{d\tau} = \underbrace{-\text{grad } \Phi(\mathbf{r})}_{\text{gewöhnlicher Teil der gravitativen Kraft}} + \underbrace{2\boldsymbol{\omega}(\mathbf{r}) \times \mathbf{v}}_{\text{gravitomagnetische Lorentzkraft}} + \mathcal{O}(v^2/c^2),$$

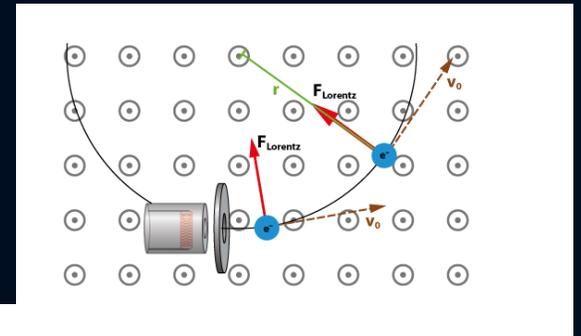
wobei  $\Phi(\mathbf{r})$  das Newtonsche Gravitationspotential und  $\mathbf{v} = (v^r, v^\theta, v^\phi)$  der Geschwindigkeitsvektor des Probekörpers ist. Die unten abgebildete Grafik zeigt die "Frame dragging" Frequenz  $\omega = \omega_z(r)$  für die Kerr Metrik, wobei bei der schwarzen Kurve  $a=0$ , bei der blauen Kurve  $a=0.99$  und bei der roten Kurve  $a=-0.99$  ist.



Die Animation veranschaulicht den allgemeinrelativistischen gravitomagnetischen Effekt eines rotierenden schwarzen Loches. Die grüne Kurve entspricht einer Situation ohne Magnetfeld (nur Coulombkraft = keine Rotation), die blaue Kurve entspricht einer Situation wo das gravitomagnetische Feld in +z-Richtung (schwarzes Loch rotiert entgegen dem Uhrzeigersinn) zeigt und bei der roten Kurve zeigt das gravitomagnetische Feld in -z-Richtung (schwarzes Loch rotiert im Uhrzeigersinn).

# Der gravitomagnetische Effekt

## Elektromagnetischer Effekt der Lorentzkraft:



## Gravito-magnetischer Effekt:

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

Diese "Frame dragging" Frequenz wirkt in ähnlicher Weise auf die Geschwindigkeit von Probekörpern, wie das Magnetfeld in der Elektrodynamik die Lorentzkraft verursacht. Im Fall 1 (grün) ist das gravitomagnetische Feld Null, im Fall 2 (blau) ist es aus der äquatorialen Ebene nach oben gerichtet und im Fall 3 zeigt es nach unten. In erster Näherung (siehe Fließbach Buch, S:172) ist die gravitomagnetische Lorentzkraft gleich  $\sim 2(\boldsymbol{\omega} \times \mathbf{v})$ , wobei  $\times$  das Kreuzprodukt,  $\boldsymbol{\omega}$  der axiale Vektor der "Frame dragging" Frequenz und  $\mathbf{v}$  der Geschwindigkeitsvektor des Probekörpers ist. Die Änderung des Geschwindigkeitsvektors nimmt in Schwachfeldnäherung dann die folgende Gestalt an:

$$\frac{d\mathbf{v}}{d\tau} = \underbrace{-\text{grad } \Phi(\mathbf{r})}_{\text{gewöhnlicher Teil der gravitativen Kraft}} + \underbrace{2\boldsymbol{\omega}(\mathbf{r}) \times \mathbf{v}}_{\text{gravitomagnetische Lorentzkraft}} + \mathcal{O}(v^2/c^2),$$

wobei  $\Phi(\mathbf{r})$  das Newtonsche Gravitationspotential und  $\mathbf{v} = (v^r, v^\theta, v^\phi)$  der Geschwindigkeitsvektor des Probekörpers ist. Die unten abgebildete Grafik zeigt die "Frame dragging" Frequenz  $\boldsymbol{\omega} = \omega_z(r)$  für die Kerr Metrik, wobei bei der schwarzen Kurve  $a=0$ , bei der blauen Kurve  $a=0.99$  und bei der roten Kurve  $a=-0.99$  ist.

# C++ Grundgerüst und Variablen

Einlesen von Header-Files  
(Definition nötiger C++  
Funktionen)

Beginn des Hauptprogramms

Ausgabe eines Strings

Deklaration einer Integer  
(natürliche Zahl) und einer  
double (reelle Zahl) Variable

Variablen bekommen einen  
festen Zahlenwert  
(Initialisierung)

Ausgabe des Wertes der  
Variablen

```
#include <iostream> //Ein-/Ausgabe (Include-Dateien)
using namespace std; //Fuer cout

main(void) //Hauptprogramm
{
    cout<<"Hello \n";
}
```



```
#include <iostream> //Ein-/Ausgabe (Include-Dateien)
using namespace std; //Fuer cout

main(void) //Hauptprogramm
{
    //Variablendeklarationen
    int i;
    double a;

    //Variableninitialisierung
    i=3;
    a=1.435553;

    cout<<"i="<<i<<"\n";
    cout<<"a="<<a<<"\n";
}
```

# Vom Quellcode zum ausführbaren Programm

Der Quellcode (z.B. Prog1.cpp) muss compiliert werden um ein ausführbares Programm (a.out) zu erzeugen. Man öffnet hierzu in dem Verzeichnis in dem sich der Quellcode befindet, ein Terminal und führt das folgende Kommando aus:

```
g++ Prog1.cpp
```

```
hاناوسكة@ITPReIAstro-Aspire-VN7-591G:~$ g++ Prog1.cpp
hاناوسكة@ITPReIAstro-Aspire-VN7-591G:~$ ./a.out
Hello
hاناوسكة@ITPReIAstro-Aspire-VN7-591G:~$ █
```

Das Programm wird gestartet und erzeugt im Terminal die Ausgabe "Hello"

Beim Compilierungsprozess wird eine Datei (a.out) erzeugt, die man dann mittels des folgenden Kommandos ausführen kann:

```
./a.out
```



# C++ Die for-Schleife

```
#include <iostream> //Ein-/Ausgabe (Include-Dateien)
using namespace std; //Fuer cout

main(void) //Hauptprogramm
{
    //Variablendeklarationen
    int i;
    double a;

    //Variableninitialisierung
    a=1.435553;

    //for Schleife
    for (i = 1; i <= 10; ++i)
    {
        cout<<"i="<<i<<"\n";
        cout<<"i mal a ="<<i*a<<"\n";
    }
}
```

Mittels einer for-Schleife können iterative Aufgaben im Programm implementiert werden. Die for-Schleife benötigt einen Anfangswert ( $i=0$ ), die Angabe wie lange sie die Iteration durchführen soll ( $i \leq 10$ ) und die Angabe um wieviel sie die Variable in jedem Schritt verändern soll ( $++i$ ). " $++i$ " bzw. " $i++$ " ist nur eine Kurzschreibweise von  $i=i+1$ .

# C++ Die do-Schleife

```
#include <iostream> //Ein-/Ausgabe (Include-Dateien)
using namespace std; //Fuer cout

main(void) //Hauptprogramm
{
    //Variablendeklarationen
    int i;
    double a;

    //Variableninitialisierung
    i=1;
    a=1.435553;

    //do Schleife
    do
    {
        cout<<"i="<<i<<"\n";
        cout<<"i mal a ="<<i*a<<"\n";
        i++;
    }
    while(i <= 10);
}
```

Mittels einer do-Schleife können iterative Aufgaben im Programm implementiert werden. Die do-Schleife benötigt lediglich eine Abbruchbedingung (while(i<=10);) wobei im Inneren der Schleife die Variable i in jedem Schritt verändert werden muss (i++). Die Variable i muss jedoch zunächst außerhalb der Schleife initialisiert werden (i=1;).

# Die TOV Gleichungen

## Allgemeine Relativitätstheorie mit dem Computer: Teil II

---

### Grundlagen zur numerischen Lösung der Tolman-Oppenheimer-Volkoff Gleichung (einfaches Euler Verfahren)

Das Differentialgleichungssystem der Tolman-Oppenheimer-Volkoff (TOV) Gleichung besitzt das folgende Aussehen

$$\frac{dp}{dr} = -\frac{(p + e)(m + 4\pi r^3 p)}{r(r - 2m)} \quad (1)$$

$$\frac{dm}{dr} = 4\pi r^2 e \quad (2)$$

$$\frac{d\phi}{dr} = \frac{m + 4\pi r^3 p}{r(r - 2m)} \quad , \quad (3)$$

wobei  $p = p(r)$  und  $e = e(r)$  der Druck und die Energiedichte der Materie darstellen,  $m = m(r)$  die radiusabhängige gravitative Masse ist und die Funktion  $\phi = \phi(r)$  die 00- bzw.  $tt$ -Komponente der Metrik bestimmt ( $g_{00} = e^{2\phi}$ ; hier bezeichnet  $e$  die Eulersche Zahl!).

# TOV-Gleichungen: Numerisches Vorgehen

Eine numerische Lösung der Sterneigenschaften benötigt lediglich Gleichung (1) und (2) und geht im einfachsten Fall (einfaches Euler Verfahren) nach folgendem Schema vor:

- Man definiert die Zustandsgleichung (EOS) der Sternmaterie als eine Funktion  $e(p)$ .
- Man startet im Sternzentrum  $r = r_0$  und legt den Wert des zentralen Druckes  $p = p_0 := p(r_0)$ , der zentralen Energiedichte  $e = e_0 := e(r_0)$  und der Masse  $m = m_0 := m(r_0) = 0$  fest. Da die TOV Gleichung (1) bei  $r_0 = 0$  singularär wird, wählt man hier einen sehr, sehr kleinen Wert für  $r_0$  (z.B.  $r_0 = 10^{-14}$ ).

$$r = 10^{-14}, \quad p = p_0, \quad e = e_0, \quad m = 0 \quad (4)$$

- Die TOV Gleichungen werden als Differenzengleichungen umgeschrieben und eine kleine Schrittweite  $dr = \Delta r \ll 1$  wird festgelegt. In einer Schleife wird dann in jedem Radiusschritt die Druck- und Massenänderung berechnet und die jeweiligen Größen beim nächsten Schritt um diesen Faktor erhöht bzw. verringert:

$$dp = - \frac{(p + e) (m + 4\pi r^3 p)}{r (r - 2m)} dr$$

- Die TOV Gleichungen werden als Differenzengleichungen umgeschrieben und eine kleine Schrittweite  $dr = \Delta r \ll 1$  wird festgelegt. In einer Schleife wird dann in jedem Radiusschritt die Druck- und Massenänderung berechnet und die jeweiligen Größen beim nächsten Schritt um diesen Faktor erhöht bzw. verringert:

$$dp = -\frac{(p + e)(m + 4\pi r^3 p)}{r(r - 2m)} dr$$

$$dm = 4\pi r^2 e dr$$

$$p = p + dp$$

$$m = m + dm$$

$$r = r + dr$$

- Im Laufe der iterativen Lösung verringert sich der Druck ständig. Die Schleife wird solange ausgeführt bis der Wert des Druckes gleich Null bzw. negativ wird (Abbruchbedingung:  $p \leq 0$ ), da an der Sternoberfläche der Druck verschwindet.

---

## TOV-Gleichungen: Numerisches Vorgehen

# C++ Lösen der TOV-Gleichung

```
#include <iostream> //Ein-/Ausgabe (Include-Dateien)
#include <math.h> //Mathematisches
using namespace std; //Fuer cout

//Definition der Zustandsgleichung
double eos(double p)
{
    double e;
    e=pow(p/10,3.0/5);
    return e;
}

main(void) //Hauptprogramm
{
    //Variablendeklarationen
    double M,p,e,r,dM,dp,de,dr;
    double eos(double);

    //Variableninitialisierung
    M=0;
    r=pow(10,-14);
    p=10*pow(0.0005,5.0/3);
    dr=0.000001;

    //do-while Schleife (Numerische Lösung der TOV-Gleichung)
    do
    {
        e=eos(p); //Wert der Energiedichte bei momentanen Druck
        dM=4*M_PI*e*r*r*dr; //Massenzunahme bei momentanem r und Schrittweite dr
        dp=- (p+e)*(M+4*M_PI*r*r*p)/(r*(r-2*M))*dr; //Druckzunahme bei momentanem r und Schrittweite dr (TOV-Gleichung)
        r=r+dr; //momentaner Radius des Neutronensterns
        M=M+dM; //momentane Masse des Neutronensterns innerhalb des Radius r
        p=p+dp; //momentaner Druck des Neutronensterns innerhalb des Radius r
    }
    while(p>0);

    //Ausgabe der Masse und des Radius auf dem Bildschirm
    cout<<"Neutronensternradius [km] = "<<r<<"\n";
    cout<<"Neutronensternmasse [Sonnenmassen] = "<<M/1.4766<<"\n";

    return 0; //main beenden (Programmende)
}
```

Die polytrope **Zustandsgleichung** ist als eine Funktion außerhalb des Hauptprogramms definiert

Deklaration der nötigen **Variablen** und der Zustandsgleichungsfunktion

Festlegung der **Anfangswerte** im Sternzentrum (M,r,p) und der Radiusschrittweite dr

**TOV-Gleichungen**

**Ausgabe auf dem Bildschirm**

# Einführung in die Parallele Programmierung

[fias.uni-frankfurt.de/~hannauske/VARTC/T2/intro/Hannauske\\_ParallelizationTut.odp](http://fias.uni-frankfurt.de/~hannauske/VARTC/T2/intro/Hannauske_ParallelizationTut.odp)

[fias.uni-frankfurt.de/~hannauske/VARTC/T2/intro/Hannauske\\_ParallelizationTut.pdf](http://fias.uni-frankfurt.de/~hannauske/VARTC/T2/intro/Hannauske_ParallelizationTut.pdf)

## Introduction

1. Parallelization on shared memory systems using OpenMP
2. Parallelization on distributed memory systems using MPI
3. Further resources

08. Juni 2018, 21.00 Uhr: Night of Science 2018, Campus Riedberg in Frankfurt am Main

# M.Hanauske: Tanz der Neutronensterne

NIGHT  
OF  
SCIENCE 2018



08.06.2018

...es wird wieder spät.