

Vorlesung

Allgemeine Relativitätstheorie mit dem Computer

PC-Pool Raum 01.120 Johann Wolfgang Goethe Universität 11. April, 2016

Matthias Hanauske

*Frankfurt Institute for Advanced Studies
Johann Wolfgang Goethe Universität
Institut für Theoretische Physik
Arbeitsgruppe Relativistische Astrophysik
D-60438 Frankfurt am Main
Germany*

Allgemeines zur Vorlesung

Ort und Zeit:

PC-Pool Raum 01.120, immer Montags von 16.00 bis 18.00 Uhr

Vorlesungs-Materialien:

<http://th.physik.uni-frankfurt.de/~hاناuske/VARTC/>

Kurs auf der Online-Lernplattform Lon Capa:

<http://lon-capa.server.uni-frankfurt.de/>

Der letzte Vorlesungstermin am Montag den 11.07.2016 muss leider auf einen anderen Termin verschoben werden.

Plan für die heutige Vorlesung:

Kurzer Überblick der Inhalte der Vorlesung

Vergabe der Login-Accounts für den PC-Pool

Kleine Einführung in Maple

Berechnung von Christoffel Symbolen und Riemann Tensor mit Maple

Vorlesung besteht aus drei Teilen

Allgemeine Relativitätstheorie mit dem Computer von Dr.phil.nat.Dr.rer.pol. [Matthias Hanauske](#)

[Home](#) [Research](#) [Contact](#)

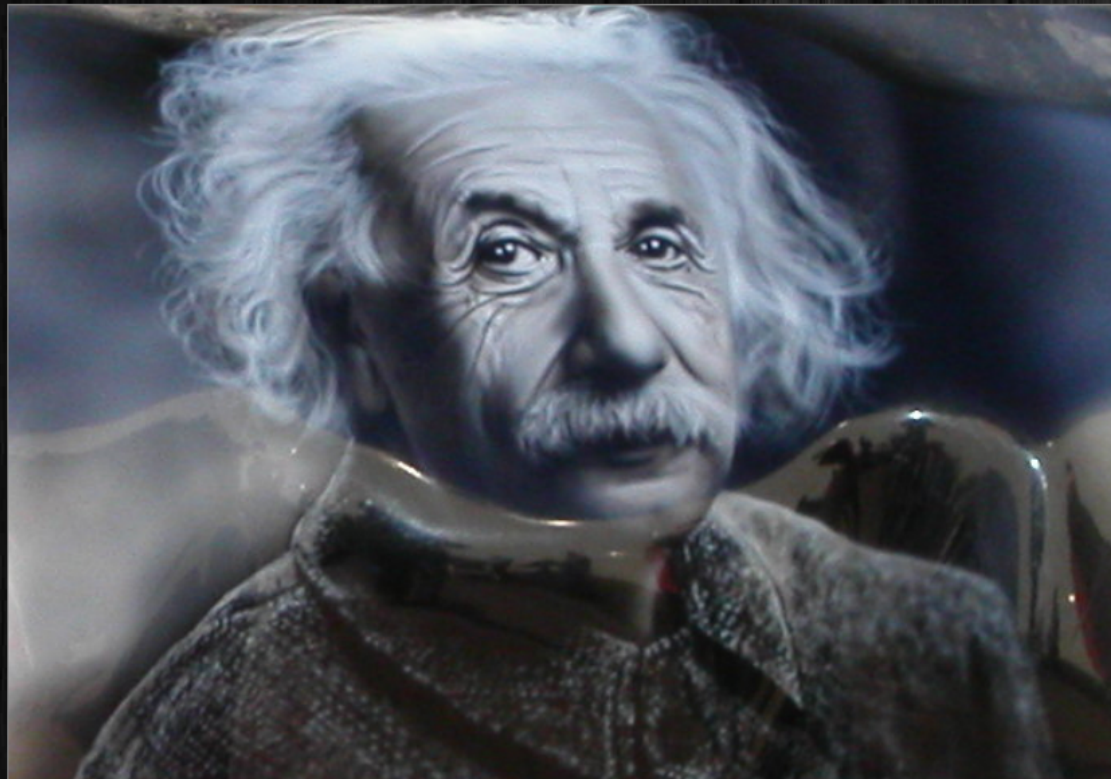
[Einführung](#)

[Teil I](#)

[Teil II](#)

[Teil III](#)

[E-Learning](#)



Allgemeine Relativitätstheorie mit dem Computer (General Theory of Relativity on the Computer) Vorlesung SS 2016, Mo. 16-18.00 Uhr, PC-Pool 01.120

In dieser Vorlesung werden die mathematisch anspruchsvollen Gleichungen der Allgemeinen Relativitätstheorie (ART) in diversen Programmierumgebungen analysiert. Im ersten Teil des Kurses erlernen die Studierenden die Verwendung von Computeralgebra-Systemen (Maple und Mathematica). Die oft komplizierten und zeitaufwendigen Berechnungen der tensoriellen Gleichungen der ART können mit Hilfe dieser Programme erleichtert werden. Diverse Anwendungen der Einstein- und Geodätengleichung werden in Maple implementiert, quasi analytische Berechnungen durchgeführt und entsprechende Lösungen berechnet und visualisiert. Der zweite Teil des Kurses befasst sich mit der numerischen Berechnung von Neutronensternen und Weißen Zwergen mittels eines C/C++ Programms. Nach einer kurzen Auffrischung der grundlegenden Programmierkenntnisse, erstellen die Studierenden, gemeinsam mit dem Betreuer, ein Programm, das die Tolman-Oppenheimer-Volkov-Gleichung numerisch löst und visualisieren die Ergebnisse. Zusätzlich wird hierbei in die Grundkonzepte der parallelen Programmierung eingeführt und eine MPI- und OpenMP-Version des C/C++ Programms erstellt. Im dritten Teil des Kurses werden zeitabhängige numerische Simulationen der ART mittels des Einstein Toolkit durchgeführt und deren Ergebnisse mittels Python/Matplotlib visualisiert. Inhaltlich wird hierbei ebenfalls auf den, dem Programm zugrunde liegenden (3+1)-Split der ART eingegangen und, abhängig von den Vorkenntnissen der Studierenden, mehrere fortgeschrittene, astrophysikalisch relevante Probleme simuliert. Mögliche Themen dieses abschließenden Teils könnten die folgenden Systeme

Teil I

Analytische Berechnungen und numerische Simulationen in Maple

Basierend auf der Einstein- und Geodätengleichung, werden im ersten Teil der Vorlesung unterschiedliche Probleme der allgemeinen Relativitätstheorie analysiert.

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

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

Nichtrotierende schwarze Löcher
Ereignishorizonte, was geschieht wenn ein Teilchen in ein schwarzes Loch fällt, Bewegung von Probekörpern und Lichtteilchen um ein schwarzes Loch,..

Eigenschaften von Neutronensternen
Verlauf der Dichte innerhalb eines NS, Masse-Radius Beziehung, Raumzeitkrümmung innerhalb und ausserhalb eines NS,..

Rotierende schwarzen Löchern
Mitführungseffekt der Raumzeit (frame-dragging),...

Teil II

Paralleles Programmieren mit C/C++ und OpenMP/MPI

Die Eigenschaften von Neutronensternen werden in diesem Teil der Vorlesung fundamental mittels der Programmiersprache C++ generiert. Ausgehend von der, im ersten Teil hergeleiteten Tollmann-Openheimer-Volkov Gleichung, wird mittels des einfachen Euler-Verfahrens die Differentialgleichung in C++ implementiert.

```
#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;

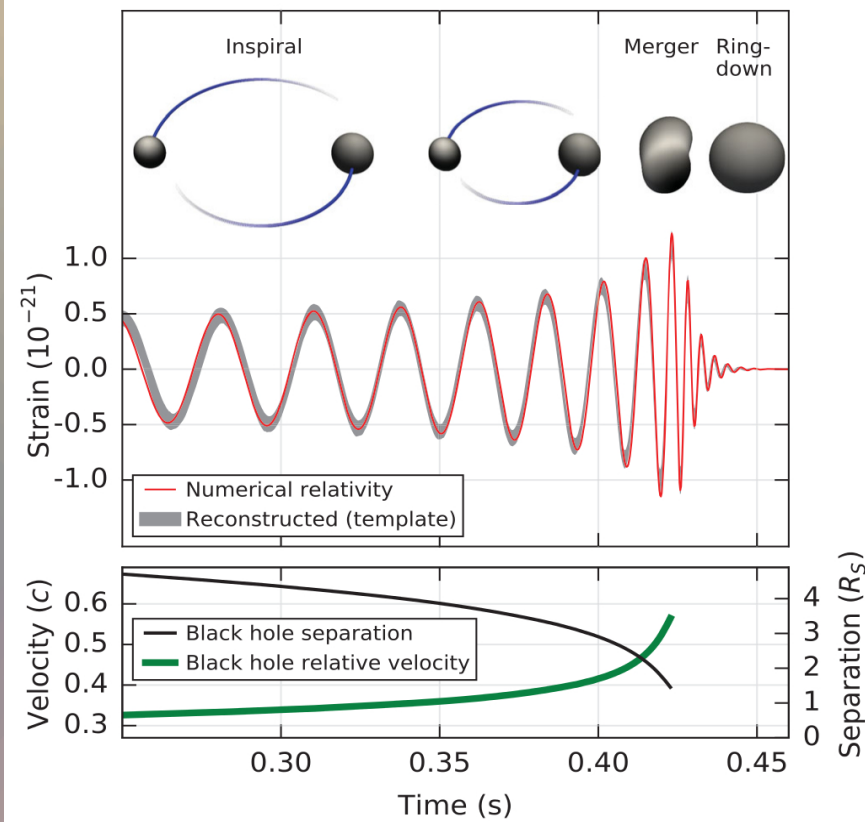
    //do-while Schleife (Numerische Lösung der TOV-Gleichung)
    do
    {
        e=eos(p); //Wert der Energiedichte bei momentanen Druck
        dM=4*M_PI*e*r*r*dr; //Massenzunahme bei momentanem r und Schrittweite dr
        dp=- (p+e)*(M+4*M_PI*r*r*p)/(r*(r-2*M))*dr; //Druckzunahme bei momentanem r und 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";

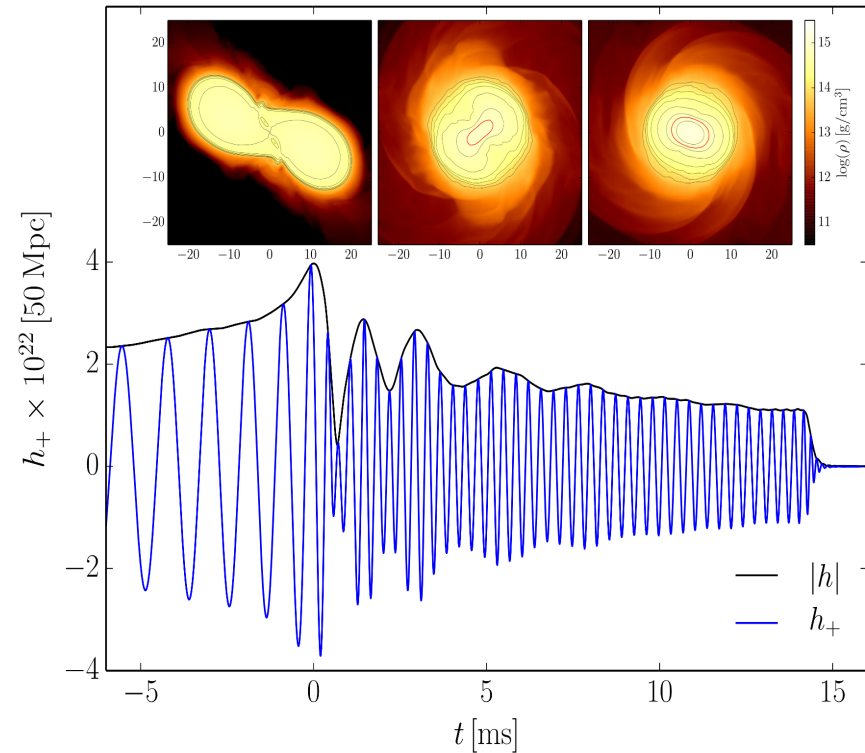
    return 0; //main beenden (Programmende)
}
```

Gravitationswellen

First observation GWs from binary black hole merger by the LIGO



Simulation of Gravitational Waves from Neutron Star Merger

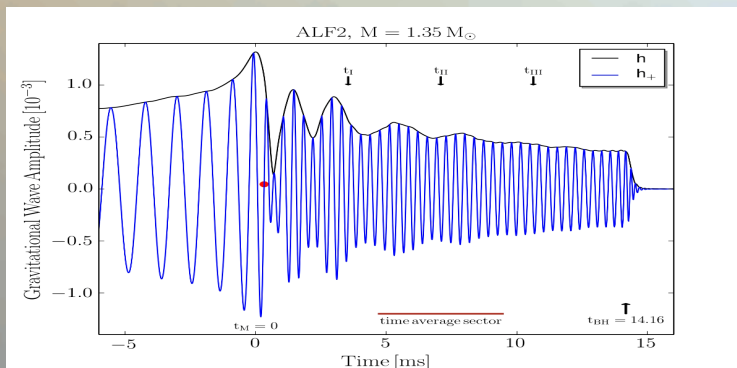


Estimated gravitational-wave strain amplitude from GW150914 (PRL 116, 06102 (2016)).

Gravitational wave amplitude h_+ and $|h|$ at a distance of 50 Mpc.

Teil III

Computersimulationen mit dem Einstein-Toolkit



In diesem Teil wird ein Einblick in allgemeinrelativistische Simulation auf Supercomputern gegeben. Unter zuhelfenahme des **Einstein-Toolkit**, werden unterschiedliche, realistische Systeme betrachtet (z.B. Neutronenstern-Kollisionen mit Aussendung von Gravitationswellen)

