

Vorlesung 7

Allgemeine Relativitätstheorie mit dem Computer

PC-Pool Raum 01.120 Johann Wolfgang Goethe Universität 30. Mai, 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

Ort und Zeit:

PC-Pool Raum 01.120, immer Montags von 16.15 bis 17.45 Uhr
Zusätzlicher, freiwilliger Übungstermin 15.00 bis 16.15 Uhr

Vorlesungs-Materialien und *Lon Capa* Online-Lernplattform:

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

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

Plan für die heutige Vorlesung:

Wiederholung: Kurze Einführung in C++.

Numerische Lösung der TOV-Gleichung in C++.

Vergleich der Resultate mit Maple.

Einführung in die parallele Programmierung mit OpenMP und MPI

Paralleles TOV Programm

Berechnung der maximalen Masse von Neutronen und Quarksternen

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";
}
```



Prog1.cpp

```
#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:

```
c++ Prog1.cpp
```

```
hanauske@ITPRelAstro-Aspire-VN7-591G:~$ c++ Prog1.cpp
hanauske@ITPRelAstro-Aspire-VN7-591G:~$ ./a.out
Hello
hanauske@ITPRelAstro-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
```



Prog1.cpp



a.out

C++ Die for-Schleife

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 ($i++$). $i++$ ist nur eine Kurzschreibweise von $i=i+1$.

```
#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";
    }
}
```

C++ Die do-Schleife

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;`).

```
#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);
}
```

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

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

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

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

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 $r_0 = 0$ singulär 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:

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

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

$$p = p + dp$$

$$m = m + dm$$

- 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 $r_0 = 0$ singulär 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 Differenzgleichungen 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:

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

- 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.
-

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);
        dM=4*M_PI*e*r*r*dr;
        dp=-(p+e)*(M+4*M_PI*r*r*r*p)/(r*(r-2*M))*dr;
        r=r+dr;
        M=M+dM;
        p=p+dp;
    }
    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

//Wert der Energiedichte bei momentanem Druck
//Massenzunahme bei momentanem r und Schrittweite dr
//Druckzunahme bei momentanem r und Schrittweite dr (TOV-Gleichung)
//momentaner Radius des Neutronensterns
//momentane Masse des Neutronensterns innerhalb des Radius r
//momentaner Druck des Neutronensterns innerhalb des Radius r

Ausgabe auf dem Bildschirm