

Realistische oder sinnvolle Physikalische Lösungen zu Albert Einsteins Feldgleichungen

Harry W.Locksley
Former Head of Mathematics
Stockton-Billingham College
Billingham, England
h.locksley@tiscali.co.uk

Übertragen aus dem Englisch ins Deutsch und etwas für das Astrophysik Forum
aufbereitet von:

Dr. W. Tenten

Die Autoren legen Wert darauf, daß dieses file nicht kommerziell ohne
ausdrückliche Genehmigung der Autoren verwendet werden darf. Die Benutzung
zu privaten Zwecken ist gestattet.

Kurzinfo

Albert Einstein (1879-1955) geboren in Ulm. Dort im Einstein Museum findet man eine
ganze Menge Material, was der Mann
so alles tat. Insbesondere 1915, wo er seine Feldtheorie veröffentlichte. Diese
Gleichungssysteme ergaben erstmals ein physikalisch
und mathematisch verwertbares Weltbild.
In diesem MAPLE file wird einmal versucht festzustellen, ob es gelingt eine "realistische"
Lösung dieser Feldgleichungen zu erhalten.

In der Einleitung wird erklärt, was hier unter "realistisch" vorzustellen ist.

einige Infoseiten im Internet:

<http://de.wikipedia.org/wiki/Energie-Impuls-Tensor>

<http://theory.gsi.de/~vanhees/faq/gravitation/node29.html>

<http://theory.gsi.de/~vanhees/faq/gravitation/node34.html>

http://www.einsteins-erben.de/Theorien/RT/Die_Einsteinschen_Feldgleichunge.htm

Suchbegriff: Die_Einsteinschen_Feldgleichunge.htm

- Einleitung

Für die Raumzeit hat Einstein diese Gleichung gefunden: $E + k \cdot g = 8 \cdot \pi \cdot T$, E ist der Einstein Tensor mit g als seiner Metrik. T ist der Energie-Impuls Tensor (Stress-Energy) und k ist die kosmologische Konstante.

Die Tensoren E , g , T können teilweise auch als 4x4 symmetrische Matrizen behandelt werden. Dann sind deren Komponenten reale differentierbare Funktionen mit ihren Raumzeitvariablen x , y , z , ct . k ist ein Skalar. Damit wird gezeigt, daß ein Tensor mit ganz "normaler" Matrizenrechnung erstellt werden kann. Die Metrik g beschreibt die Raumzeitbeziehung und muß in jedem Raumzeitpunkt die Signatur 2 besitzen.. Sie muß 3 positive und einen negativen Eigenwert(e) haben.

Der Einstein Tensor E hängt von der Metrik g ab. Diese Abhängigkeiten werden mit Hilfe der Christoffel Symbole, des Riemann und des Ricci Tensors sowie des Ricci skalars beschrieben. Damit bestimmt die Wahl von g und k den Energie-Impuls Tensor und in dessen Folge die Lösung der Feldgleichungen.

Es ist natürlich sehr schwer zu sagen, welche Lösungen richtig oder falsch sind. Ich drücke das mal so aus: Eine interpretierbare Lösung ist mit einer höheren Wahrscheinlichkeit "richtiger" als eine nicht-interpretierbare. Die Aussage "richtig" "falsch" kann zur Zeit nicht bewertet werden. Dazu fehlen nachwievor vielzuviele Messungen.

Sollte eine der Lösungen die Energie Bedingungen von Hawking und Ellis nicht erfüllen, wird er als "nicht physikalische Realität" gekennzeichnet.

Diese Bedingungen besagen, daß an jedem Raumzeitpunkt x , y , z , ct der Tensor T einen nicht-negativen Eigenwert $e[0]$ -das ist dann eine Energie Dichte- besitzt, welcher die anderen drei Eigenwerte $e[1]$, $e[2]$, $e[3]$ -den prinzipiellen Energie-Impuls Tensor- in absoluten Werten dominiert.

$0 \leq e[0]$ and $-e[0] \leq e[j] \leq e[0]$ ($j = 1..3$) für alle x , y , z , ct

Wenn T real symmetrisch ist, dann muss T immer 4 reale Eigenwerte besitzen .

- Die Lösungs Methodik

Die diagonale Metrik g wird wie folgt geschrieben:

$g[i,i] = s \cdot \exp[s \cdot \{(a + \cos w)^2 + b\}^p]$ ($i = 1..4$) mit: $w = d \cdot (t + x + y + z)$ and s, a, b, p und

d als Konstanten gegeben als:

for i = 1 : s = -1, a = 0.5, b = 1, p = -1, d = 1*10⁻⁸
for i = 2 : s = 1, a = -0.5, b = 4, p = 3, d = 1*10⁻⁸
for i = 3 : s = 1, a = 1, b = 4, p = 3, d = 1*10⁻⁸
for i = 4 : s = 1, a = -1, b = 4, p = 3, d = 1*10⁻⁸.

Der Energie-Impuls Tensor wird bei $k = 1 \cdot 10^{-9}$ durch seine Metrik g definiert und kann als $\cos(w)$ und $\sin(w)$ (Quadratur Niederschrift) mit Hilfe der symbolischen Analytik gefunden werden. Maxima und Minima jeder Komponente von T werden innerhalb des Rechenintervalls $0 \leq w \leq 2 \cdot \pi$ mit der Schrittweite $1e-4$ berechnet.

Die nicht-diagonal Elemente der Matrix T haben beide (obere und untere Dreiecksmatrix) Maxima und Minima der Größenordnung $\leq 1e-13$.

Die Maxima Minima von $T[i,j]$ gemäß unten durchgeführter berechnung zeigen:

	Minimum	Maximum
T[1,1]	-10 ⁻¹¹	-10 ⁻¹¹
T[2,2]	+10 ¹⁷	+10 ⁹⁵
T[3,3]	+10 ¹⁷	+10 ²¹¹
T[4,4]	+10 ¹⁷	+10 ²¹¹

Die Ergebnisse zeigen, daß T in jedem Punkt 3 sehr große Eigenwerte besitzt ($> 1e17!!!$) und einen kleinen negativen $< 1e-11$.

Damit ist die dominante Energiebedingung (siehe oben) erfüllt.

Ähnlich realistische lösungen können auch erhalten werden indem man die konstanten Werte von k und d nachjustiert.

$k = 1e-20$ $d = 1e-j$ mit $j = 13 \dots 19$. Diese Berechnungen werden mit 60 digits Genauigkeit sowie einer Schrittweite von $1e-3$ durchgeführt

Anschliessende Prüfung mit 100 Digits und $1e-4$ Schrittweite!

Diese Lösung wird erreicht durch die Empfindlichkeitsfunktion von T respektive der Variation metrischer Parameter für verschiedene Prüfmetriken.

Das bedingt eine sehr große Menge an Lösungen und an Vergleichen!

Mit numerischer Methodik können die maxima und Minima von T berechnet werden, jedoch sind diese nicht theoretisch beschreibbar.

Jedoch kann ein gewisses Vertrauen in diese Lösungen gelegt werden, wenn man die Sequenzen im Intervall $0 \leq w \leq 2 \cdot \pi$ und deutlich kleinerer Schrittweite z.B. $1e-6$ durchführt. Das jedoch verlängert die rechenzeit ungemein!!!

Die folgenden Berechnungen mit der oben angegebenen Genauigkeit sowie Schrittweite sind

denen der Genauigkeit 100 Digits und der kleineren Schrittweite $1e-4$ sehr gut vergleichbar.

Die Konstanten k und d werden mit $u[0]$ und $u[j]$ $j=1..4$ angegeben, damit die Lösungen, für die $k=1e-20$ mit $d = 1e-13$

$u[0] = 1e-20$ $u[j]=1e-13$, $j=1..4$ erreicht werden können. Das sind realistische Größen, für welche w eine ganz gute linear Kombination der Raumzeit ist.

Der erste Teil der Berechnung beschäftigt sich mit der symbolischen Ermittlung des Energie Impuls Tensors T , der hier "str" genannt wird..

Die Metrik g wird m genannt.

Der Einstein Tensor E wird im Programm "et" genannt und die kosmologische Konstante k wird als $k[0]$ bezeichnet.

$T(\text{str})$ ist symmetrisch und somit brauchen nur 10 seiner 16 Komponenten numerisch ermittelt werden.

Diese sind: "ff[1] bis ff[10] mit ihren Diagonal Elementen ff[1], ff[5], ff[8}, ff[10]

- Berechnung

- Initialisierung

```
> restart:

n:=4:
Digits:=60:

seq(k[i],i=0..n):
seq(u[i],i=0..n):
seq(x[i],i=1..4):

m:=array(1..4,1..4):

im:=array(1..4,1..4):

for pp from 1 to 4 do
  for q from 1 to 4 do
    m[pp,q]:=0:im[pp,q]:=0:
  end do:
end do:
```

- Die Metrik Diagonal Komponenten

```
> m[1,1]:= -exp(-
((0.5+cos(k[1]*x[1]+k[2]*x[2]+k[3]*x[3]+k[4]*x[4]))^2+1)^(
-1)):

m[2,2]:=exp(
((-0.5+cos(k[1]*x[1]+k[2]*x[2]+k[3]*x[3]+k[4]*x[4]))^2+4)^
```

```

(3)):

m[3,3]:=exp(
((1+cos(k[1]*x[1]+k[2]*x[2]+k[3]*x[3]+k[4]*x[4]))^2+4)^(3)
):

m[4,4]:=exp(
((-1+cos(k[1]*x[1]+k[2]*x[2]+k[3]*x[3]+k[4]*x[4]))^2+4)^(3)
)):

for i from 1 to 4 do:
    im[i,i]:=1/m[i,i]:
end do:

```

- Lösen nach Christoffel Elemente: ch4

```

> ch1:=array(1..4,1..4,1..4):ch2:=array(1..4,1..4,1..4):

ch3:=array(1..4,1..4,1..4):ch4:=array(1..4,1..4,1..4):

for pp from 1 to 4 do          # erst einmal alle Matrixelemente zu Null
setzen
    for q from 1 to 4 do
        for r from 1 to 4 do

ch1[pp,q,r]:=0:ch2[pp,q,r]:=0:ch3[pp,q,r]:=0:ch4[pp,q,r]:=
0:

                end do:
            end do:
        end do:

for pp from 1 to 4 do
    for q from 1 to 4 do      # Eintrag der Christoffelelemente
        for r from 1 to 4 do
            for s from 1 to 4 do:

                ch1[pp,q,r]:=ch1[pp,q,r] +
im[s,pp]*diff(m[s,q],x[r]):

                ch2[pp,q,r]:=ch2[pp,q,r] +
im[s,pp]*diff(m[s,r],x[q]):

                ch3[pp,q,r]:=ch3[pp,q,r] +
im[s,pp]*diff(m[q,r],x[s]):

                    end do:
                end do:
            end do:
        end do:
    end do:

for pp from 1 to 4 do      # und jetzt wird der Tensor bestimmt

```

```

        for q from 1 to 4 do
            for r from 1 to 4 do
                ch4[pp,q,r]:=0.5*(ch1[pp,q,r] +
ch2[pp,q,r] - ch3[pp,q,r]):
            end do:
        end do:
    end do:
end do:

```

- Lösen nach Riemann Tensor: rm5 (gekrümmte Koordinaten des 3-d Raums)

```

> rm1:=array(1..4,1..4,1..4,1..4):rm2:=array(1..4,1..4,1..4,
1..4):
rm3:=array(1..4,1..4,1..4,1..4):rm4:=array(1..4,1..4,1..4,
1..4):
rm5:=array(1..4,1..4,1..4,1..4):

for pp from 1 to 4 do:
    for q from 1 to 4 do:      # wie oben: erst Matrix Elemente
zu Null setzen
        for r from 1 to 4 do:
            for s from 1 to 4 do:
                rm3[pp,q,r,s]:=0:
                rm4[pp,q,r,s]:=0:
            end do:
        end do:
    end do:
end do:

for pp from 1 to 4 do
    for q from 1 to 4 do
        for r from 1 to 4 do
            for s from 1 to 4 do

rm1[pp,q,r,s]:=diff(ch4[pp,q,s],x[r]):

rm2[pp,q,r,s]:=diff(ch4[pp,q,r],x[s]):
            end do:
        end do:
    end do:
end do:

for pp from 1 to 4 do
    for q from 1 to 4 do
        for r from 1 to 4 do
            for s from 1 to 4 do
                for ww from 1 to 4 do

```

```

rm3[pp,q,r,s]:=rm3[pp,q,r,s] + ch4[pp,ww,r]*ch4[ww,q,s]:
rm4[pp,q,r,s]:=rm4[pp,q,r,s] + ch4[pp,ww,s]*ch4[ww,q,r]:
        end do:
        end do:
        end do:
end do:

for pp from 1 to 4 do          # und schliesslich der Riemann Tensor
    for q from 1 to 4 do
        for r from 1 to 4 do
            for s from 1 to 4 do

rm5[pp,q,r,s]:=rm1[pp,q,r,s] - rm2[pp,q,r,s] +
rm3[pp,q,r,s] - rm4[pp,q,r,s]:
                end do:
            end do:
        end do:
    end do:
end do:

```

- Lösen nach Ricci Tensor ric (gekrümmte Koordinaten der Raumzeit)

```

> ric:=array[1..4,1..4]:

for pp from 1 to 4 do:
    for q from 1 to 4 do
        ric[pp,q]:=0:
    end do:
end do:

for pp from 1 to 4 do:
    for q from 1 to 4 do:
        for r from 1 to 4 do:
            ric[pp,q]:=ric[pp,q] +
rm5[r,pp,r,q]:
        end do:
    end do:
end do:

```

- Lösen nach Ricci Skalar rs (Krümmungsskalar)

```

> rs:=im[1,1]*ric[1,1] + im[2,2]*ric[2,2] + im[3,3]*ric[3,3]
+im[4,4]*ric[4,4]:

```

- Lösen nach Einstein Tensor et

```

> et:=array(1..4,1..4):

for pp from 1 to 4 do
    for q from 1 to 4 do
        et[pp,q]:=ric[pp,q] - 0.5*rs*m[pp,q]:
    end do:
end do:

```

- Lösen nach Energie-Impuls Tensor str

```

> str:=array(1..4,1..4):
for pp from 1 to 4 do
    for q from 1 to 4 do
        str[pp,q]:=et[pp,q] +k[0]*m[pp,q]:
        str[pp,q]:=str[pp,q]/(8*Pi):
    end do:
end do:

```

- subst. w for (k[1]* x[1]+k[2]* x[2]+k[3]* x[3]+k[4]* x[4]) in str

```

> for pp from 1 to 4 do
    for q from 1 to 4 do
        feg:=str[pp,q]:

geg:=subs({k[1]*x[1]+k[2]*x[2]+k[3]*x[3]+k[4]*x[4]
=w},feg):

        str[pp,q]:=geg:
    end do:
end do:

ff:=array(1..10):
ff[1]:=str[1,1]:ff[2]:=str[2,1]:ff[3]:=str[3,1]:ff[4]:=str
[4,1]:ff[5]:=str[2,2]:
ff[6]:=str[3,2]:ff[7]:=str[4,2]:ff[8]:=str[3,3]:ff[9]:=str
[4,3]:ff[10]:=str[4,4]:

```

- Konstanten

```

> u[0]:=1*10^(-9):
u[1]:=1*10^(-8):u[2]:=1*10^(-8):u[3]:=1*10^(-8):u[4]:=1*10
^(-8):

```

- Lösen nach max & min Werten von T mit Schrittweite h

```

> h:=0.001:
a11:=array(-100..6400):a7:=array(-100..6400):

for rr from 1 to 10 do:

```

```

print("rr = ",evalf(rr)):

a8:=ff[rr]:
a9:=unapply(a8,w,seq(k[i],i=0..n)):

for i from -100 to 6400 do:
    a12:=i*h:
    a13:=a9(a12,seq(u[i],i=0..n)):
    a7[i]:=evalf(-a13):
    a11[i]:=evalf(a13):
end do:

a14:=max(seq(a11[i],i=-100..6400)):
a15:=max(seq(a7[i],i=-100..6400)):
print(evalf(a14)):
print(evalf(-a15)):

print("*****"):

end do:

"rr = ", 1.

-0.146181233547063412192660152574045896955235047456859976196421 10-10
-0.292503501270153073908686133382860447667157973356006302417154 10-10
*****

"rr = ", 2.

0.152788745368219522338128412837613787553081259910838190797761 10-14
-0.115208338362941139934418702115349068671662615731144542895054 10-12
*****

"rr = ", 3.

0.361414242482690750322221688003248633141254150831370250976359 10-14
-0.152692916517236735017215507354233897682621810776860722917128 10-12
*****

"rr = ", 4.

0.24285246326407270279797702730788093199559635547044897323100 10-13
-0.126947270784874570478198108146739835255944937350931108806974 10-12
*****

```

"rr = ", 5.

0.425315127716980908473500924485806051162857121510242460304893 10^{96}

0.247257727354381180813794510150748820334910869500127473674168 10^{18}

"*****"

"rr = ", 6.

0.152910724442852606194851345827457022930698178819292626591692 10^{-14}

-0.743104224205165548886901568929127756572715736030363839062702 10^{-13}

"*****"

"rr = ", 7.

0.152845250081743272161478312398574443539602369844237169418618 10^{-14}

-0.121668400325480228624175969781777346577595234167992398377740 10^{-12}

"*****"

"rr = ", 8.

0.908942635819648102993937329269327402020066451320113346796390 10^{212}

0.248104084011681907583287313296778724445398864943971195018158 10^{18}

"*****"

"rr = ", 9.

0.271265504909752701887709976868384019457568339922623471654705 10^{-13}

-0.425723832768362986411630927625638518479640828845815464987488 10^{-14}

"*****"

"rr = ", 10.

0.908916927529551793392346299954136341473451156387441655235514 10^{212}

0.248100142290937361054072207648033883373395211477385156349587 10^{18}

"*****"

Zusammenfassung

Die hier angewandte Methode, bei der physikalisch sinnvolle Lösungen der Einstein Feldgleichungen gefunden werden, zeigte die Fähigkeit mit modernen Rechnerhilfsmitteln solche Lösungen zu finden. Das war zur Zeit Einsteins einfach nicht möglich. Trotzdem war Einstein auf dem richtigen Weg, jedoch verrechnete sich der gute Mann einigermassen häufig, was ihm bis heute von einigen Leuten Zweifel an seinen Theorien anhängen läßt.

Diese Zweifel sind jedoch -zumindest aus jetziger Sicht- völlig belanglos:

Denn Einsteins Methodik war einfach gesagt: GENIAL!!!

Referenzen

[1] : "A first course in general relativity" by Bernard F. Schutz (Cambridge University Press, 1996)

[2] : "Introducing Einstein's Relativity" by Ray d'Inverno (Clarendon Press, Oxford, 1996).