Organisation der Übungen: **Antje Peters** **Raum 2.107** **peters@th.physik.uni-frankfurt.de**

Blatt 12

vom 20.1.2017, Abgabe am 27.1.2017 in der Vorlesung

39) Viererstromdichte (mündlich) *(5+1=6 Punkte)*

Zeige exemplarisch, dass sich $j^\mu(x) = (c\rho(\mathbf{r},t), \mathbf{j}(\mathbf{r},t))$ unter Lorentz-Transformationen wie ein Vierervektor verhält, d.h. dass die Vergabe eines oberen griechischen Index μ an j gerechtfertigt ist. Betrachte dazu

i. eine im Inertialsystem Σ ruhende homogen geladene Kugel (Ladungsdichte ρ_0),

ii. einen im Inertialsystem Σ auf der x-Achse liegenden homogen geladenen Draht (Linienladungsdichte λ_0, verschwindender Strom).

Das Inertialsystem Σ' bewegt sich relativ zu Σ in x-Richtung mit konstanter Geschwindigkeit v. Gib für die beiden genannten Fälle sowohl j^μ als auch j'^μ an (verwende für j'^μ Dein Wissen über Längenkontraktion) und zeige danach, dass j^μ und j'^μ über eine entsprechende Lorentz-Transformation verbunden sind.

40) Elektrodynamik in d Raumdimensionen (schriftlich) *(2+2+2=6 Punkte)*

Betrachte die Lagrange-Dichte der Elektrodynamik

$$\mathcal{L} = -\frac{1}{16\pi} F^{\mu\nu} F_{\mu\nu} - \frac{1}{c} j^\mu A_\mu$$

in $d \geq 1$ Raumdimensionen $\mathbf{r} = (x^1, x^2, \ldots, x^d)$.

i. Bestimme die Feldgleichungen in kovarianter Form, d.h. in Komponentenschreibweise mit Vierindizes

ii. Wie viele Komponenten hat das elektrische Feld in d Dimensionen? Wie lauten die Feldgleichungen der Elektrostatik in d Raumdimensionen, die die Ladungsdichte $\rho(\mathbf{r})$ mit dem Potential $A^0(\mathbf{r}) = \Phi(\mathbf{r})$ beziehungsweise den Komponenten des elektrischen Feldes in Verbindung bringen?

iii. Wie viele Komponenten hat das magnetische Feld in d Dimensionen? Wie lauten die Feldgleichungen der Magnetostatik in d Raumdimensionen, die die Stromdichte $\mathbf{j}(\mathbf{r})$ mit dem Potential $\mathbf{A}(\mathbf{r})$ beziehungsweise den Komponenten des magnetischen Feldes in Verbindung bringen?

41) Fourier-Transformation (schriftlich) ***(3+3+2=8 Punkte)***

In der Vorlesung wurde die Fourier-Transformation diskutiert und dabei die Konvention

$$f(x) = \frac{1}{\sqrt{2\pi}} \int dk\, e^{+ikx} \tilde{f}(k) \quad , \quad \tilde{f}(k) = \frac{1}{\sqrt{2\pi}} \int dx\, e^{-ikx} f(x) \qquad (\star)$$

verwendet.

i. Berechne die Fourier-Transformierte $\tilde{f}(k)$ einer normierten Gauß-Funktion

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}.$$

Diskutiere anhand geeigneter Skizzen, welchen Einfluss die Breite σ auf das Ergebnis $\tilde{f}(k)$ hat.

ii. Berechne die beiden Integrale

$$\frac{1}{\sqrt{2\pi}} \int dk\, e^{+ikx} e^{-ikx_0} \quad , \quad \frac{1}{\sqrt{2\pi}} \int dx\, e^{-ikx} \delta(x - x_0)$$

und zeige damit exemplarisch die Gültigkeit von ($\star$), d.h. dass "Fourier-Hin-Transformation" und "Fourier-Rück-Transformation" invers zueinander sind.
Hinweis: Bei der Berechnung des ersten Integrals ist es zweckmäßig, den Integranden mit $e^{-\epsilon|k|}$ zu multiplizieren ($\epsilon > 0$ und klein) und am Ende der Rechnung den Limes $\epsilon \to 0$ zu bilden.

iii. Zeige, dass die Fourier-Transformierte einer reellen symmetrischen Funktion eine reelle symmetrische Funktion ist. Zeige, dass die Fourier-Transformierte einer reellen antisymmetrischen Funktion eine rein imaginäre antisymmetrische Funktion ist.

Python-Übung 12 – Rechnen mit Matrizen und Vektoren am Beispiel von Vierervektoren

Es kann praktisch sein, Rechnungen mit Vierervektoren mit Hilfe von Python zu überprüfen. `numpy`-Arrays bieten u.a. die Möglichkeit, Vierervektoren und 4x4-Matrizen zu implementieren und Operationen wie die Matrixmultiplikation auszuführen.

Der angegebene Code erzeugt Vierervektoren „mit oberen und unteren Indizes“, sowie die Metrik η und eine Rotationsmatrix Λ_x, die eine spezielle Lorentz-Transformation ist. Außerdem werden Matrixmultiplikationen mit `dot(..., ...)` ausgeführt und die Ergebnisse ausgegeben.

Starte den Code auf deinem Computer und sieh dir die Ausgabe an, die er produziert.

1. Manipuliere den Code so, dass statt der Rotationsmatrix Λ_x eine Matrix $\Lambda_{x\text{-boost}}$ implementiert ist, die einen Lorentz-Boost in x-Richtung ausführt.

2. Manipuliere den Code so, dass er die folgenden Rechenoperationen ausführt (mit $X'_\mu = (\Lambda_{x\text{-boost}})_\mu{}^\beta X_\beta$):

   - $X_\mu X^\mu$
   - $X'_\mu X'^\mu$
   - $X_\mu X_\mu$
   - $X'_\mu X'_\mu$

   (Beachte, dass die letzten beiden Operationen im Rahmen einer kovarianten Schreibweise nicht zu Lorentz-invarianten Größen führen.)

3. Welche Rechenoperationen liefern dieselben Ergebnisse und warum?

Code:

```
from numpy import *

Eta = diag([+1, -1, -1, -1])
print "Eta="
print Eta

def Lambda_x(Phi):
   # Rotationsmatrix um Winkel Phi
   return [
    [1,0,0,0],
    [0,1,0,0],
    [0,0,cos(Phi), -sin(Phi)],
    [0,0,sin(Phi),  cos(Phi)]
   ]

# Kovarianter Vierervektor
Xu = array([1,2,3,4])
print "Xu=", Xu

# Kontravarianter Vierervektor durch Matrixmultiplikation
Xo = dot(Eta, Xu)
print "Xo=", Xo

# rotierter kovarianter Vierervektor
XuStrich = dot( Lambda_x(pi/2), Xu)
print "Lambda_x(pi/2)="
print Lambda_x(pi/2)
print "XuStrich=", XuStrich
```