Röntgenfluoreszenz

In [3]:
```
#import modules
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.stats import chi2
%matplotlib inline
plt.rcParams["figure.figsize"][0] = 14
plt.rcParams["figure.figsize"][1] = 9
plt.rcParams['errorbar.capsize']=2
```

In [10]:
```
#fit functions
#mosleys law
def mosley(x, Er_sqrt, sig12):
    return Er_sqrt*(x-sig12)*np.sqrt(1/n1**2-1/n2**2)
```

Messung verschiedener Elemente

In [38]:
```
#measured data
#Elements in the order Fe, Ti, Ag, Mo, Ni, Zn, Cu, Zr
Z = np.array([26, 22, 47, 42, 28, 30, 29, 40]) #atomic number
K_alpha = np.array([6.41, 4.43, 21.96, 17.48, 7.50, 8.68, 8.06, 15.81]) #energy o
K_alpha_err = np.array([15, 17, 17, 17, 16, 16, 16, 16])*1e-2*0.25
K_alpha_sqrt = np.sqrt(K_alpha)
K_alpha_sqrt_err = 0.5*K_alpha_err/K_alpha_sqrt
```

In [39]:
```
#fit mosley function to mesured data
n1 = 1 #fuction parameters for K_alpha
n2 = 2
popt, pcov = curve_fit(mosley, Z, K_alpha_sqrt, sigma=K_alpha_sqrt_err)
perr = np.sqrt(np.diag(pcov))

#show fit parameters
print('Er_sqrt', popt[0], '+-', perr[0])
print('sig12', popt[0], '+-', perr[0])
```

```
Er_sqrt 0.118793285672 +- 0.00028723702113
sig12 0.118793285672 +- 0.00028723702113
```

In [40]:
```
#fit quality
chi_squared1=np.sum((mosley(Z,*popt)-K_alpha_sqrt)**2/K_alpha_sqrt_err**2)
dof1=Z.size-2
chi_squared_red1=chi_squared1/dof1
print('Wir erhalten die nachfolgenden Werte für die Güte des Fits:')
print('chi_squared= ' + str(chi_squared1))
print('chi_squared_red= ' + str(chi_squared_red1))
print()

prob1=round(1-chi2.cdf(chi_squared1,dof1),2)*100
print('Die Fitwahrscheinlichkeit beträgt: ' + str(prob1) + ' %')
```

```
Wir erhalten die nachfolgenden Werte für die Güte des Fits:
chi_squared= 5.37129572177
chi_squared_red= 0.895215953629

Die Fitwahrscheinlichkeit beträgt: 50.0 %
```

In [41]:
```
#plot data and fit fuction
x = np.linspace(20, 50, 20)
plt.figure('Energie_sqrt')
plt.errorbar(Z, K_alpha_sqrt, K_alpha_sqrt_err, label = 'Messwerte', linestyle =
plt.plot(x, mosley(x, *popt), label = 'Fit')
plt.xlabel('Kernladungszahl Z')
plt.ylabel(r'$\sqrt{E_\alpha}/\sqrt{keV}', fontsize=14)
plt.title(r'$\sqrt{E_\alpha}$ als Funktion von Z')
plt.legend(frameon = True)
plt.savefig('Diagramme/Energie.pdf')
```