NMR

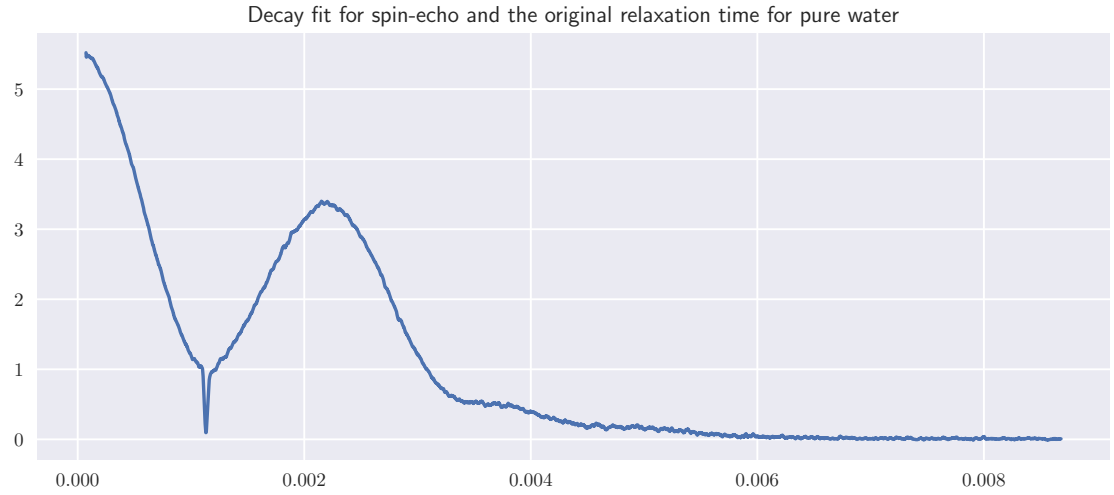
October 23, 2023

[1]:
```
import numpy as np
import pandas as pd
import sympy as sm
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from matplotlib_inline.backend_inline import set_matplotlib_formats
plt.style.use('seaborn-v0_8')
plt.rcParams |= {
    'text.usetex': True,
    'figure.figsize': (10, 4)
}
set_matplotlib_formats('svg', 'pdf')
```

[2]:
```
def smooth_data(data, window_size):
    return data.rolling(window=window_size, min_periods=1).mean()

# Function for the decay exponential
def decay_func(t, A, k, C):
    return A * np.exp(-k * t) + C

t = sm.symbols('t')
def decay_func_sm(t, A, k, C):
    return A * sm.exp(-k * t) + C
```

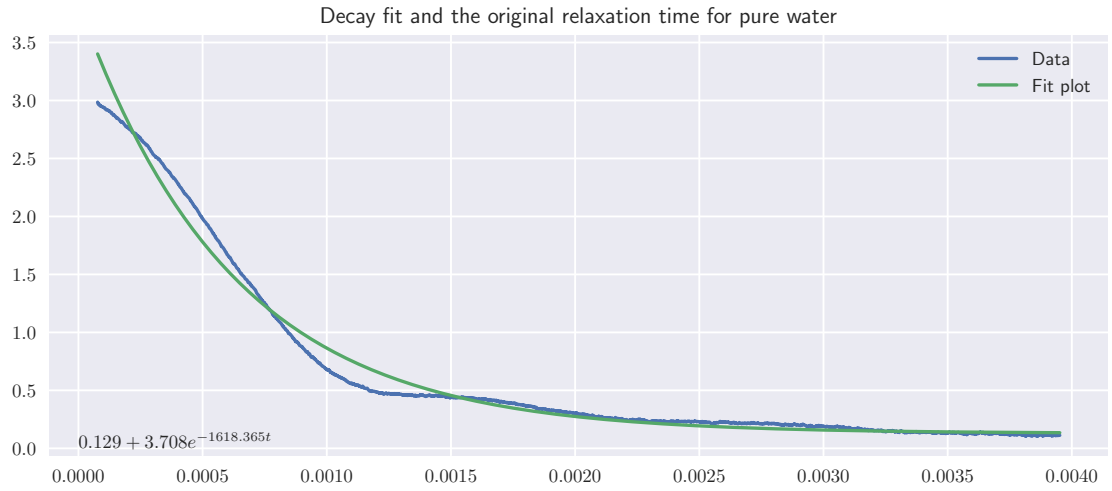
[15]:
```
df = pd.read_csv('/Users/ammar-imac/Documents/NMR/tek0004CH1H_OSE.csv')
df = df[19:]
df.columns = ['TIME', 'CH1']
df.TIME = df.TIME.astype(float)
df.CH1 = df.CH1.astype(float)
```

[17]:
```
val = np.argmax(df.CH1)
plt.plot(df.TIME.iloc[val:], smooth_data(df.CH1.iloc[val:], 20))
plt.title("Decay fit for spin-echo and the original relaxation time for pure␣
 ↪water")
plt.show()
```



[5]:
```
df = pd.read_csv('/Users/ammar-imac/Downloads/NMR/tek0008ALLLLLLL.csv')
df = df[19:]
df.columns = ['TIME', 'CH1']
df.TIME = df.TIME.astype(float)
df.CH1 = df.CH1.astype(float)
# fit plot
val = np.argmax(df.CH1)
t_data = df.TIME.iloc[val:]
y_data = smooth_data(df.CH1.iloc[val:], 10)
params, covariance = curve_fit(decay_func, t_data, y_data)
```

[6]:
```
plt.plot(df.TIME.iloc[val:], smooth_data(df.CH1.iloc[val:], 10), label='Data')
plt.plot(
    df.TIME.iloc[val:], decay_func(df.TIME.iloc[val:].astype(float), *params),
    label='Fit plot'
)
plt.text(0, 0, sm.latex(decay_func_sm(t, *[round(val, 3) for val in params]),␣
 ↪mode='inline'))
plt.title("Decay fit and the original relaxation time for pure water")
plt.legend()
plt.show()
```

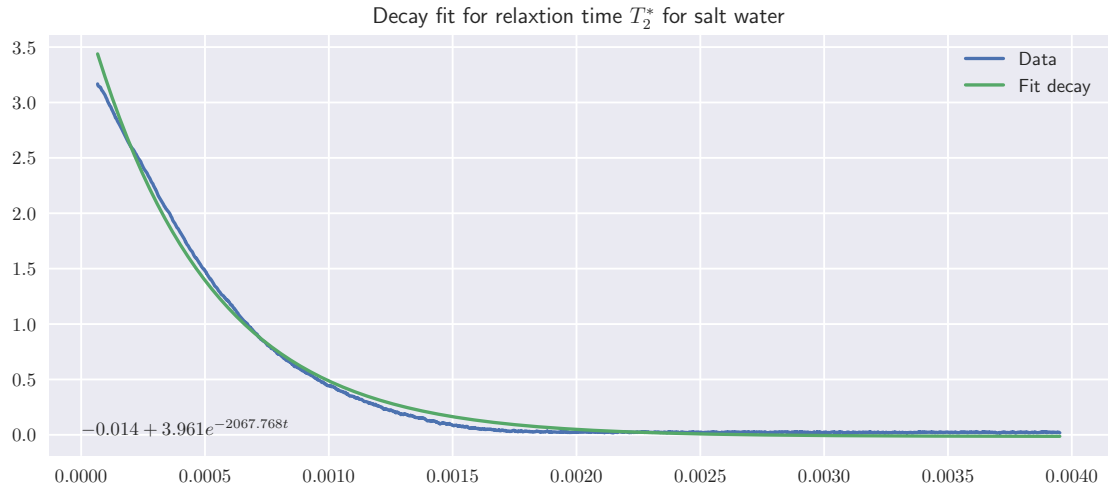


[7]:
```
df = pd.read_csv('/Users/ammar-imac/Documents/NMR/tek0001CH1.csv')
df = df[19:]
df.columns = ['TIME', 'CH1']
df.TIME = df.TIME.astype(float)
df.CH1 = df.CH1.astype(float)
# fit plot
val = np.argmax(df.CH1)
t_data = df.TIME.iloc[val:]
y_data = smooth_data(df.CH1.iloc[val:], 10)
params, covariance = curve_fit(decay_func, t_data, y_data)
```

[8]:
```
val = np.argmax(df.CH1)
plt.plot(df.TIME.iloc[val:], smooth_data(df.CH1.iloc[val:], 10), label='Data')
plt.plot(
    df.TIME.iloc[val:], decay_func(df.TIME.iloc[val:].astype(float), *params),
    label='Fit plot'
)
plt.text(0, 0, sm.latex(decay_func_sm(t, *[round(val, 3) for val in params]),␣
 ↪mode='inline'))
plt.title("Decay fit relaxation tiem $T_2^*$ and the original relaxation time␣
 ↪for heavy oil")
plt.legend()
plt.show()
```

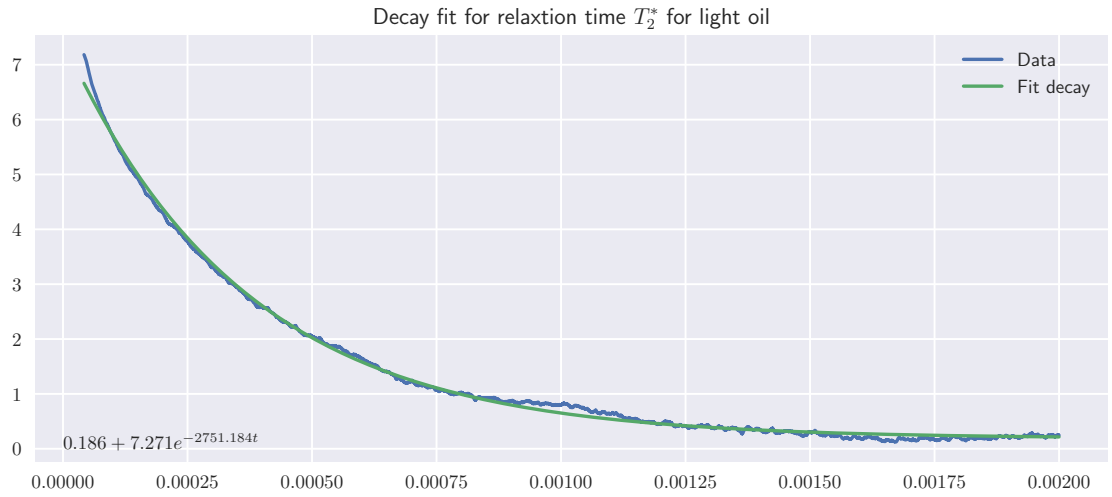
[9]:
```
df = pd.read_csv('/Users/ammar-imac/Documents/NMR/tek0009CHWSSS1.csv')
df = df[19:]
df.columns = ['TIME', 'CH1']
df.TIME = df.TIME.astype(float)
df.CH1 = df.CH1.astype(float)
# fit plot
val = np.argmax(df.CH1)
t_data = df.TIME.iloc[val:]
y_data = smooth_data(df.CH1.iloc[val:], 10)
params, covariance = curve_fit(decay_func, t_data, y_data)
```

[10]:
```
val = np.argmax(df.CH1)
plt.plot(df.TIME.iloc[val:], smooth_data(df.CH1.iloc[val:], 10),
        label='Data')
plt.plot(
    df.TIME.iloc[val:], decay_func(df.TIME.iloc[val:].astype(float), *params),
    label='Fit decay'
)
plt.text(0, 0, sm.latex(decay_func_sm(t, *[round(val, 3) for val in params]),␣
 ↪mode='inline'))
plt.title("Decay fit for relaxtion time $T_2^*$ for salt water")
plt.legend()
plt.show()
```



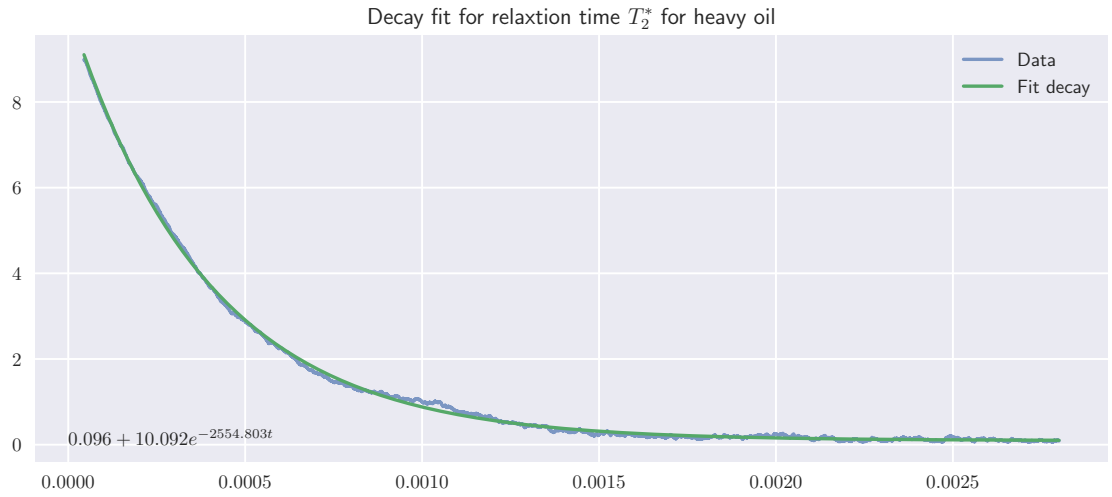
[11]:
```
df = pd.read_csv('/Users/ammar-imac/Documents/NMR/2nd/tek0000CH1.csv')
df = df[19:]
df.columns = ['TIME', 'CH1']
df.TIME = df.TIME.astype(float)
df.CH1 = df.CH1.astype(float)
# fit plot
val = np.argmax(df.CH1)
t_data = df.TIME.iloc[val:]
y_data = smooth_data(df.CH1.iloc[val:], 10)
params, covariance = curve_fit(decay_func, t_data, y_data)
```

[12]:
```
val = np.argmax(df.CH1)
plt.plot(df.TIME.iloc[val:], smooth_data(df.CH1.iloc[val:], 10),
        label='Data')
plt.plot(
    df.TIME.iloc[val:], decay_func(df.TIME.iloc[val:].astype(float), *params),
    label='Fit decay'
)
plt.text(0, 0, sm.latex(decay_func_sm(t, *[round(val, 3) for val in params]),␣
 ↪mode='inline'))
plt.title("Decay fit for relaxtion time $T_2^*$ for light oil")
plt.legend()
plt.show()
```



[13]:
```
df = pd.read_csv('/Users/ammar-imac/Documents/NMR/2nd/tek0002.csv')
df = df[19:]
df.columns = ['TIME', 'CH1']
df.TIME = df.TIME.astype(float)
df.CH1 = df.CH1.astype(float)
# fit plot
val = np.argmax(df.CH1)
t_data = df.TIME.iloc[val:]
y_data = smooth_data(df.CH1.iloc[val:], 10)
params, covariance = curve_fit(decay_func, t_data, y_data)
```

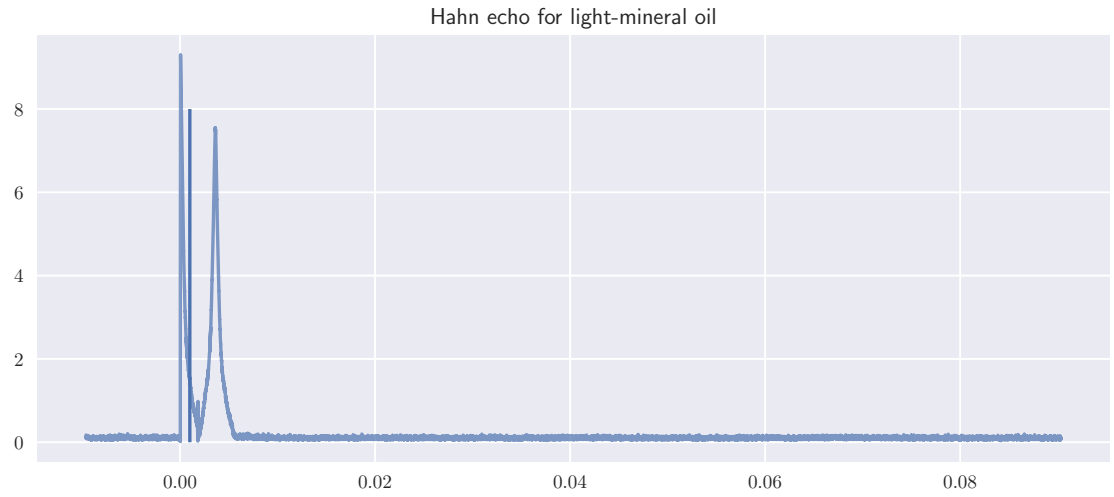
[14]:
```
val = np.argmax(df.CH1)
plt.plot(df.TIME.iloc[val:], smooth_data(df.CH1.iloc[val:], 10),
         label='Data', alpha=0.7)
plt.plot(
    df.TIME.iloc[val:], decay_func(df.TIME.iloc[val:].astype(float), *params),
    label='Fit decay'
)
plt.text(0, 0, sm.latex(decay_func_sm(t, *[round(val, 3) for val in params]),␣
 ↪mode='inline'))
plt.title("Decay fit for relaxtion time $T_2^*$ for heavy oil")
plt.legend()
plt.show()
```



[ ]:

[172]:
```
dfs = []
for i in range(3, 8 + 1):
    n = str(i).zfill(2)
    df = pd.read_csv(f'/Users/ammar-imac/Documents/NMR/2nd/tek00{n}.csv',␣
 ↪low_memory=False)
    df = df[19:]
    df.columns = ['TIME', 'CH1']
    df.TIME = df.TIME.astype(float)
    df.CH1 = df.CH1.astype(float)
    dfs.append(df)
```

[173]:
```
df = dfs[5]
plt.plot(df.TIME, smooth_data(df.CH1, 10), label='Data', alpha=0.7)
plt.title("Hahn echo for light-mineral oil")
lim = 0.001
plt.vlines(lim, 0, 8)
plt.show()
```

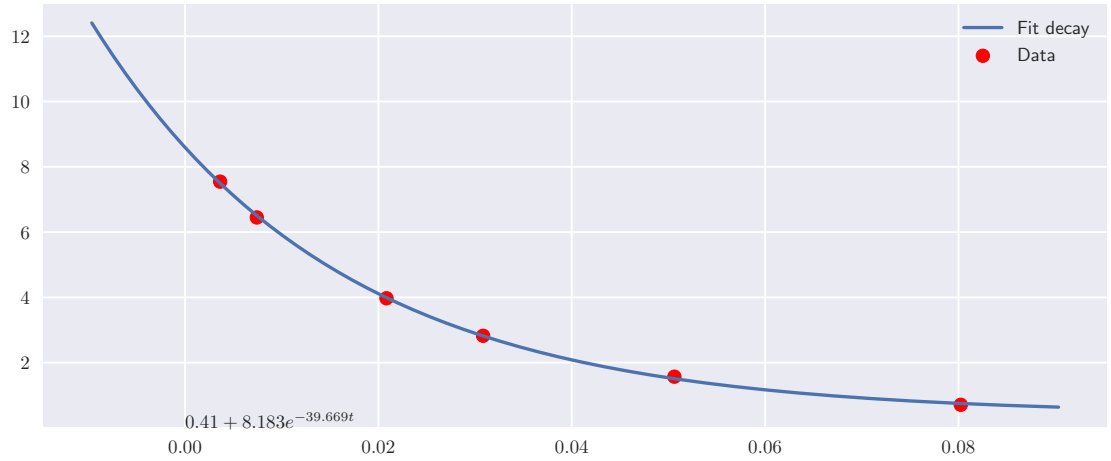


[145]:
```
idx = df.CH1[df.TIME > lim].idxmax()
df.iloc[idx]
```

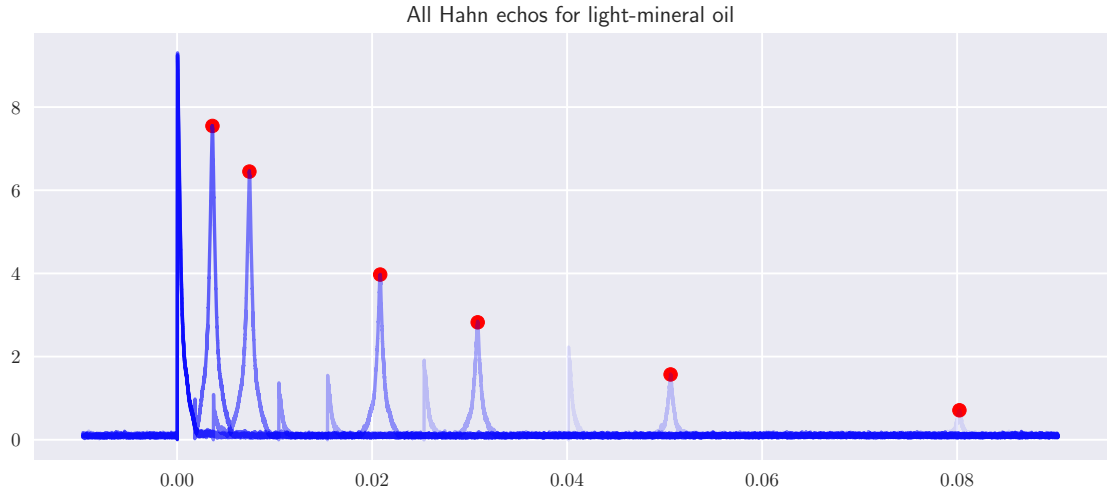
[145]:
```
TIME    0.003627
CH1     7.546260
Name: 132790, dtype: float64
```

[175]:
```
# points of max Hahn echo for light oil
time = [0.080236, 0.050613, 0.030829, 0.020832, 0.007418, 0.003627]
data = [0.709131, 1.570780, 2.824080, 3.972590, 6.449100, 7.546260]
# fit decay
params, covariance = curve_fit(decay_func, time, data)
```

[184]:
```
plt.text(0, 0, sm.latex(decay_func_sm(t, *[round(val, 3) for val in params]),␣
 ↪mode='inline'))
plt.plot(df.TIME, decay_func(df.TIME, *params),label='Fit decay')
plt.scatter(time, data, c='r', label='Data')
plt.legend()
plt.show()
```



[181]:
```
c = 0
for df in dfs:
    c += 1
    plt.plot(df.TIME, smooth_data(df.CH1, 10), label='Data', alpha=c/10, c='b')
plt.scatter(time, data, c='r')
plt.title("All Hahn echos for light-mineral oil")
plt.show()
```

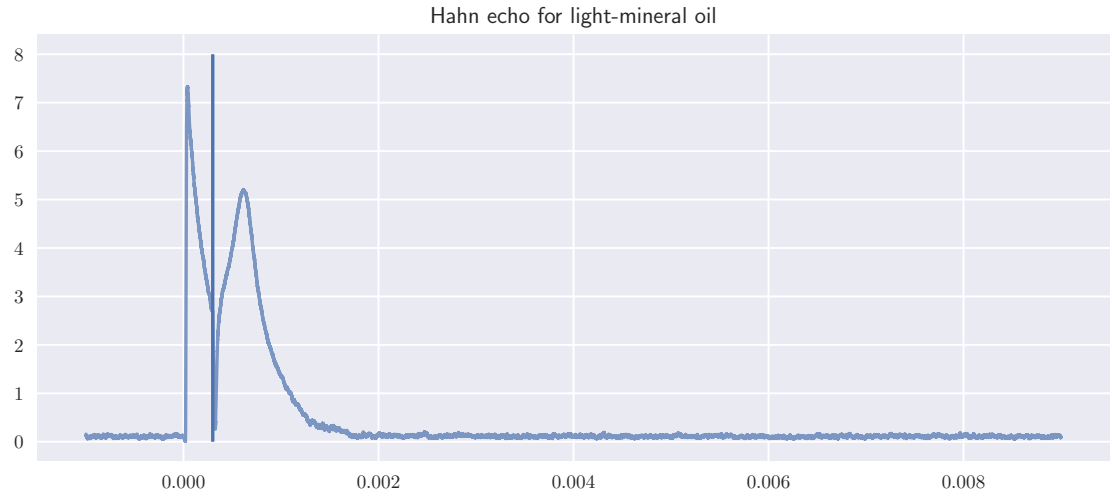


[ ]:
```

```

[188]:
```
dfs = []
for i in range(9, 13 + 1):
    n = str(i).zfill(2)
```

```
    df = pd.read_csv(f'/Users/ammar-imac/Documents/NMR/2nd/tek00{n}.csv',␣
↪low_memory=False)
    df = df[19:]
    df.columns = ['TIME', 'CH1']
    df.TIME = df.TIME.astype(float)
    df.CH1 = df.CH1.astype(float)
    dfs.append(df)
```

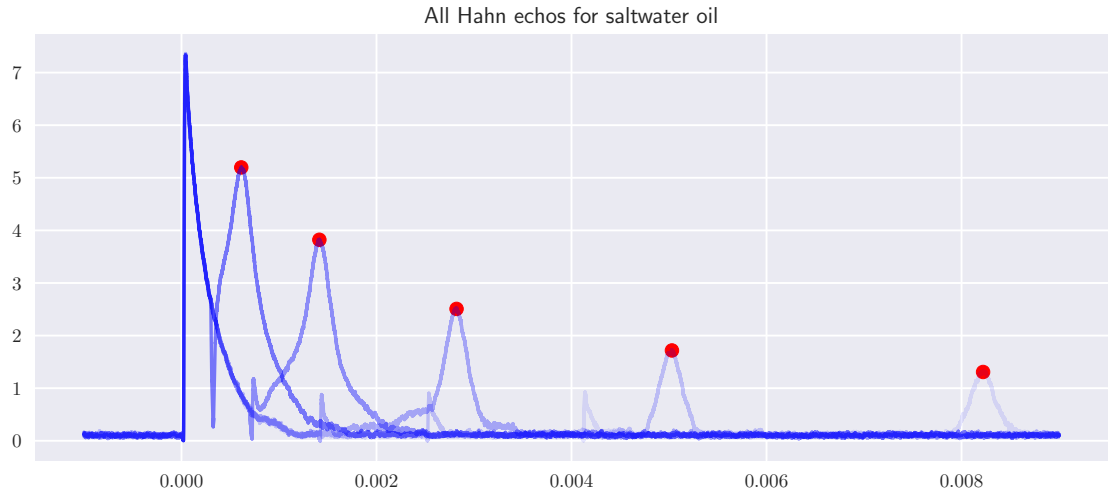
[169]:
```
df = dfs[4]
plt.plot(df.TIME, smooth_data(df.CH1, 10), label='Data', alpha=0.7)
plt.title("Hahn echo for light-mineral oil")
lim = 0.0003
plt.vlines(lim, 0, 8)
plt.show()

idx = df.CH1[df.TIME > lim].idxmax()
df.iloc[idx]
```



[169]:
```
TIME    0.000613
CH1     5.197190
Name: 161276, dtype: float64
```

[189]:
```
# points of max Hahn echo for saltwater
time = [0.008221, 0.00503, 0.002821, 0.001414, 0.000613]
data = [1.308120, 1.71750, 2.507500, 3.821870, 5.197190]
# fit decay
params, covariance = curve_fit(decay_func, time, data)
```

[190]:
```
c = 0
for df in dfs:
    c += 1
    plt.plot(df.TIME, smooth_data(df.CH1, 10), label='Data', alpha=c/10, c='b')
plt.scatter(time, data, c='r')
plt.title("All Hahn echos for saltwater oil")
plt.show()
```



[193]:
```
plt.text(0, 2, sm.latex(decay_func_sm(t, *[round(val, 3) for val in params]),␣
 ↪mode='inline'))
plt.plot(df.TIME, decay_func(df.TIME, *params),label='Fit decay')
plt.scatter(time, data, c='r', label='Data')
plt.legend()
plt.show()
```