



Nr.: FIN-02-2014

Cointegration Analysis of Financial Time Series Data

Johannes Steffen, Pascal Held, Rudolf Kruse

Arbeitsgruppe Computational Intelligence



Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg

Technical report

Nr.: FIN-02-2014

Cointegration Analysis of Financial Time Series Data

Johannes Steffen, Pascal Held, Rudolf Kruse

Arbeitsgruppe Computational Intelligence

Technical report (Internet)
Elektronische Zeitschriftenreihe
der Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg
ISSN 1869-5078



Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg

Impressum (§ 5 TMG)

Herausgeber:

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Der Dekan

Verantwortlich für diese Ausgabe:

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Pascal Held
Postfach 4120
39016 Magdeburg
E-Mail: pascal.held@ovgu.de

http://www.cs.uni-magdeburg.de/Technical_reports.html

Technical report (Internet)
ISSN 1869-5078

Redaktionsschluss: 26.04.2014

Bezug: Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Dekanat

Technical Report: Cointegration Analysis of Financial Time Series Data

Johannes Steffen, Pascal Held, and Rudolf Kruse

Otto von Guericke University Magdeburg
Department of Knowledge and Language Engineering

Table of Contents

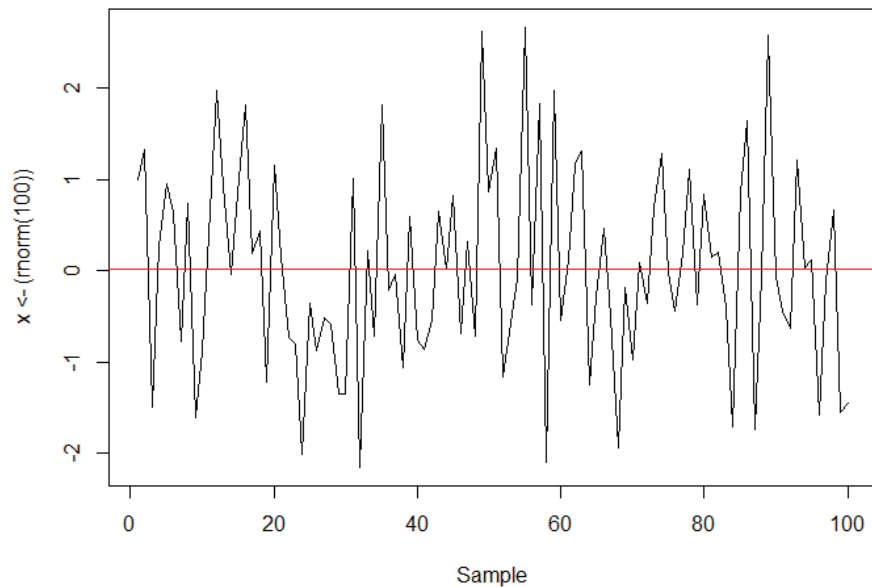
Introduction	
1	Mean Reversion 1
Theory and Concepts	
2	Testing for Stationarity 4
2.1	Augmented Dickey Fuller Test 4
3	Cointegrating Time Series 6
3.1	Covariate Augmented Dickey Fuller Test 6
3.2	Johansen Test 7
4	Further Measurements and Tests 10
4.1	Hurst Exponent 10
4.2	Variance Ratio Test 11
4.3	Half Life of Mean Reversion 12
Application on Foreign Exchange Markets	
5	Introduction 14
6	A Strategy for FX Markets 14
6.1	Data, Data Processing, and Time Period 14
6.2	A Basic Strategy 16
6.3	Sorting Criteria for Choosing a Pair to Trade 17
6.4	Entries and Exits 17
6.5	When to Stop Trading 19
6.6	Forward Test Results 21
7	Improvements and Further Considerations 21

Introduction

1 Mean Reversion

The characteristic of a *mean reverting* time series is that, once deviated from its mean value, it will tend to revert to it. In other words, the change of its consecutively following period is proportional to the mean value and the previously obtained value in a mean-reverting time series. Though a Gaussian distribution is not necessarily assumed in general, Figure 1 shows a simple example of a mean reverting time series constructed from 100 random samples drawn from a normal population:

Fig. 1: Constructed mean-reverting time series using samples (X) from a normal population ($X \sim N(0, 1)$)

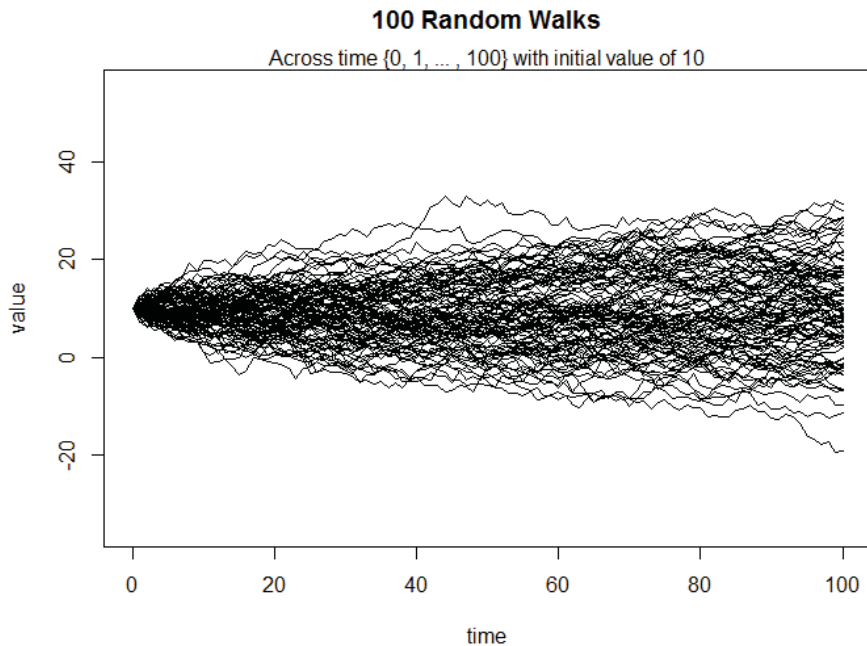


This artificially generated time series is mean-reverting in the strongest way as linear mean drifts (trending behaviours throughout the time series) are absent. Hence, this time series is not only reverting to its mean but is also *mean stationary*, thus, it has a homogeneous mean across all observations. However, a time series may also be mean-reverting w.r.t. a heterogeneous mean.

Mathematically we speak of a stationary time series if the variance of the logarithmized data increase (much) slower than it would be the case for a random walk (a path/series connecting randomly drawn data points). Thus, the variance of a random walk series increases linearly w.r.t. to time and the variance of a stationary series is a sub-linear function, hence, with an exponent less than 1.

This function could be approximated by τ^{2H} , where τ is the time between to samples being drawn and H the Hurst Exponent (see subsection 4.1). From the above it follows that the Hurst Exponent H has to be less than 0.5 for a slower than "normal" increasing variance (e.g. a random walk). Examples of 100 random walks (thus with $H \approx 0.5$) can be seen in Figure 2.

Fig. 2: 100 random walk each containing 101 samples in discrete time intervals.



Throughout this report, several tests and statistics will be introduced to measure the aforementioned properties of stationarity. Therefore, an appropriate way of estimating the Hurst Exponent will be discussed as well as a test to see whether the Null-hypothesis of $H = 0.5$ (the series has random walk characteristics) can be rejected and at what levels of (un)certainty (Variance Ratio Test - (Lo and MacKinlay, 2001)). Furthermore, a more advanced test, namely the **Augmented Dickey Fuller** test (Dickey and Fuller, 1979) will be described to test for stationarity.

For the remainder of this report we will exclusively focus on financial data when speaking of time series', thus, data points/ values of a time series are equivalent to price data. Unfortunately, there are almost no stationary price series' in the financial markets as most of them follow somewhat a random walk - a topic that is widely discussed in economics and econometric theory in the context of the efficient market theory. However, there may exist a relation between n ($n \geq 2$) price series resulting in a *new* constructed price series in which the Null-hypothesis of non-stationarity can be rejected. This (linear) price data combination of n different time series into one price data series is called *cointegration* and the resulting price series w.r.t. financial data is called a *cointegrated pair*. Due to practicability w.r.t. actual trading using cointegrated pairs (see chapter 4.3) we will reduce ourselves to linear combinations of price data to construct such cointegrated pairs. Therefore, two statistical tests, the **Covariate Augmented Dickey Fuller** test (**CADF**) for two time series and the Johansen test (Johansen, 1991) for an arbitrary amount of time series', are described in 3.2 to find out those coefficients necessary for constructing a cointegrated pair if possible at all.

In the second part of this report the aforementioned methods and concepts will be applied to develop a practical strategy to trade on the Foreign Exchange Markets. Therefore, we will introduce the basic software written in R and will evaluate a simple backward test within a period ranging from 2008 till the end of 2013.

Theory and Concepts

2 Testing for Stationarity

Now that we introduced the basics of mean reversion and stationary it is important to elaborate on certain criteria helping us to test for stationarity and additionally for the degree or certainty of stationarity. we will begin by introducing a simple statistical test to identify stationarity of a price series, namely, the Augmented Dickey Fuller test described in the following subsection.

2.1 Augmented Dickey Fuller Test

The Augmented Dickey Fuller Test (**ADF**) (Dickey and Fuller, 1979) is used to determine (or respectively test) whether a price series is indeed stationary or not. If the former is the case then we would expect a dependency between formerly obtained prices and future prices, thus, if the previously obtained price was above the mean we would expect (with a probability strictly > 0.5) that the upcoming price event will tend to go down towards the mean and, analogously, if the the previous price is less than the mean, that the upcoming price will go up towards the mean.

The change of the prices while observing a price series can be written as

$$\Delta y = \lambda y(t-1) + \mu + \beta t + \alpha_1 \Delta y(t-1) + \dots + \alpha_k \Delta y(t-k) + \varepsilon_t \quad (1)$$

where $\Delta y(t) \equiv y(t) - y(t-1)$.

As can be seen in eq. 1 the overall aim of the ADF-test is do determine whether the hypothesis of $\lambda = 0$ can be rejected or not (assuming that the price's change is described linearly like above). Thus, if the Null-hypothesis of $\lambda = 0$ cannot be rejected we would conclude with some specific certainty that the price changes (new incoming data points) are completely independent ($y(t) \perp\!\!\!\perp y(t-1)$) w.r.t previously obtained data, hence, we would conclude that the series follows a random walk as described in section 1.

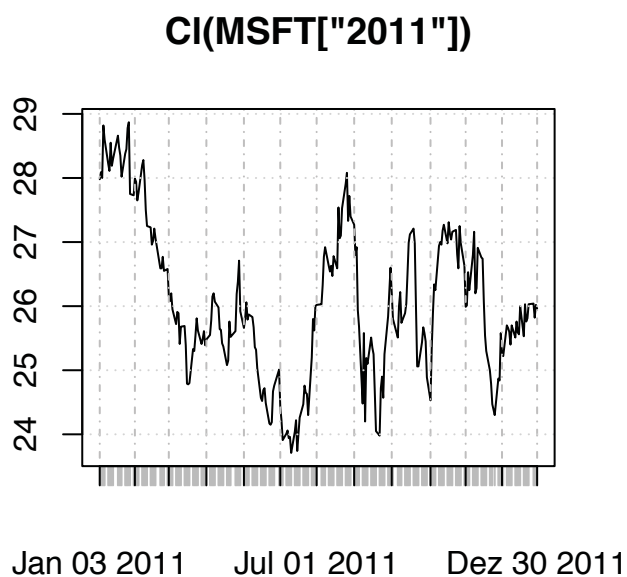
In the work of (Dickey and Fuller, 1979) the test statistics (the regression coefficient λ as the independent variable and $\Delta y(t)$ as the dependent variable divided by the standard error (SE) of the regression), the corresponding distribution, and critical values where found and explained in detail. Therefore, we may use the findings directly to check, with what amount of certainty we can reject the Null-hypothesis ($\lambda = 0$), and thus, whether the price series follows a random walk or not. Though the ADF-test was implemented in C++ due to performance gains it is also part of the R Package "tseries".

As an example we will download daily prices of the MSFT (Microsoft) stock and chart the year 2011 as can be seen below:

```
require(quantmod)
require(tseries)
getSymbols("MSFT", src = "yahoo")

## [1] "MSFT"

# Plot overview (MSFT daily closings)
plot(Cl(MSFT["2011"]))
```



The chart of the price series indicates that the series consists of both, mean reverting and trending parts. E.g., the first quarter of the year seems to indicate a trending downward move while the rest appears to go sideways and, thus, mean reverting. We will therefore calculate the p-value of the ADF-test using R for both subsets.

```
# A downwards trending part of the chart and its ADF-test
# results
adf.test(Cl(MSFT["2011-01/2011-04"]),
         alternative="stationary", k=2)
```

```

##
## Augmented Dickey-Fuller Test
##
## data: Cl(MSFT["2011-01/2011-04"])
## Dickey-Fuller = -1.423, Lag order = 2, p-value = 0.8122
## alternative hypothesis: stationary

# A stationary part of the chart and its ADF-test results
adf.test(Cl(MSFT["2011-05/2011-12"]),
         alternative="stationary", k=2)

##
## Augmented Dickey-Fuller Test
##
## data: Cl(MSFT["2011-05/2011-12"])
## Dickey-Fuller = -3.033, Lag order = 2, p-value = 0.1452
## alternative hypothesis: stationary

```

As can be seen in the test results above there is indeed evidence that the MSFT stock followed a random walk from 2011-01 to 2011-04 and contrarily had mean reverting behaviour from 2011-05 to 2011-12.

3 Cointegrating Time Series

Now that we introduced a test statistics to evaluate whether we have random walk or mean reverting behaviour we will focus on the concept of cointegration. Unfortunately, not many price series are intrinsically mean reverting by definition. Usually, most price series' will follow a random walk or consist of alternating trending and sideways periods. To cope with this rareness of mean reverting stocks and price series we will try to construct a stationary pair based on linear combinations of multiple non-stationary pairs. Thus, we combine multiple price series' that are not stationary to a new synthetic price series that will be stationary. Two methods of doing this are described in the following two subsections.

3.1 Covariate Augmented Dickey Fuller Test

The Covariate Augmented Dickey Fuller Test is used to combine *two* non-stationary price series to form an artificial cointegrated and stationary price series linearly. The idea behind it is quite intuitive: First, an ordinary linear regression (with one price

series as the dependent and the other as the independent variable) is performed to fit the two price series. Afterwards, we apply the normal Augmented Dickey Fuller test as described in subsection 2.1 to test for stationarity. The method was firstly proposed by (Engle and Granger, 1987) and is widely used in econometric applications. Since this approach is (in its simplest form) dependent on the choice of the dependent variable and is only applicable when working with two price series we will not go into more details. However, since we would like to deal with an arbitrary amount of input series' the Johansen test is more appropriate for our analysis and is further elaborated in the following subsection.

3.2 Johansen Test

Since we want to form a stationary price series out of an arbitrary number of price series' we can rewrite equation 1 using vectors and matrices for the data and there coefficients:

$$\Delta Y(t) = \Lambda Y(t-1) + M + A_1 \Delta Y(t-1) + \dots + A_k \Delta Y(t-k) + \varepsilon_t \quad (2)$$

Thus, Λ and A are matrices and $Y(t)$ is in vector representation. Similar to the argumentation in subsection 2.1 (see also (Chan, 2013)) we need to test whether our Null-hypothesis that $\Lambda = 0$ (hence, no dependence between previously obtained and recent data and therefore no cointegration) can be rejected and at what certainty (Johansen, 1991).

To actually find the linear coefficients for constructing a stationary price series a Eigenvalue decomposition of Λ is applied. Let r be the rank of Λ ($r = \text{rank}(\Lambda)$) and n the number of price series' to be cointegrated than the Johansen test (Johansen, 1991) performs an Eigenvalue decomposition of Λ and test whether we can reject the n null-hypothesis' $r \leq 0, r \leq 1, r \leq 2, \dots$, and $r \leq n-1$. Thus, for an arbitrary positive integer n (the number of price series to be cointegrated) the Johansen test calculates the test statistics for the null-hypothesis' that $r \leq i$; where $i = 0, 1, \dots, n-1$. If all $n-1$ null hypothesis are rejected within a high degree of confidence (e.g., $\alpha = 0.05$) we may conclude that indeed $r = n$. Therefore, the Johansen test borrows the idea from the PCA (principle component analysis) and transforms our data using linear combination to a new coordinate system. Furthermore, since we basically applied PCA we now may use the eigenvector corresponding to the highest eigenvalue for our hedge ratios. Critical values for the test statistic results can be found in (Johansen, 1991).

To demonstrate the application of the Johansen test we choose three currency pairs (GBPUSD, NZDUSD, USDCAD) normalized to a pip-value (a PIP is the unit that determines price changes in the foreign exchange market, usually, for all but JPY (Japanese

YEN) it is the 4th decimal point and for JPY-pairs it is the 2nd). The detailed steps are explained in the source code below:

```
require(xts) # for time series
require(quantmod) # for plotting
require(urca) # for johansen test

## Loading required package: urca

# cdata was copied from Metatrader 4 and is in matrix format
load("./data/cdata")
# show first 6 data points as an example
head(cdata)

##                GBPUSD NZDUSD USDCAD
## 2013-10-02 02:00:00 16189  8269  10332
## 2013-10-02 03:00:00 16195  8274  10328
## 2013-10-02 04:00:00 16191  8256  10337
## 2013-10-02 05:00:00 16167  8251  10338
## 2013-10-02 06:00:00 16164  8227  10345
## 2013-10-02 07:00:00 16177  8209  10340

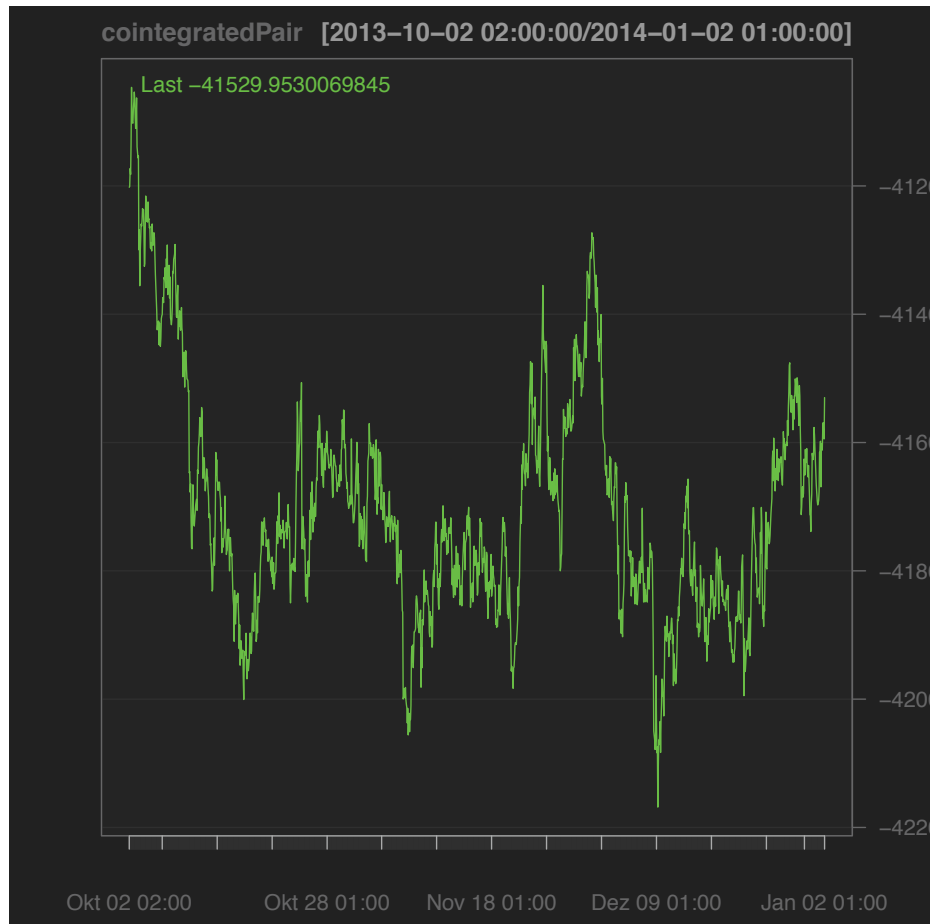
# perform johansen test from package urca
result <- ca.jo(x=cdata, type="eigen", ecdet="const", K=2)
# result summary print out
summary(result)@teststat

## [1]  1.919 10.522 16.214

summary(result)@cval

##           10pct  5pct  1pct
## r <= 2 |  7.52  9.24 12.97
## r <= 1 | 13.75 15.67 20.20
## r = 0  | 19.77 22.00 26.81

# store coefficients (first eigenvector)
coeffs <- vector()
coeffs[1:3] <- result@V[1:3]
# construct and plot synthetic pair (cointegrated)
cointegratedPair <- as.xts(cdata %*% coeffs, index(cdata))
chartSeries(cointegratedPair)
```



As can be seen when comparing the test statistics

```
summary(result)@teststat
## [1] 1.919 10.522 16.214
```

with the critical values provided by (Johansen, 1991)

```
summary(result)@cval
##          10pct  5pct  1pct
## r <= 2 |  7.52  9.24 12.97
## r <= 1 | 13.75 15.67 20.20
## r = 0  | 19.77 22.00 26.81
```

since $1.919 < 7.52$ we know that the originally time series may not be mean reverting at hand. Secondly, $10.522 < 13.75$ tells us that there also is no good linear combination of only two of the three currency pairs to form a stationary pair. In this case, even a linear combination of all three pairs might not result in truly stationary pair, since $16.214 < 19.77$ (at least our confidence is not very high, thus, strictly below the 10pct). However, even if we can not guarantee stationary with high confidence ($\leq 10pct$) we still might be able to profitable trade this pair as the chart looks promising.

Therefore, more features have to be calculated on the created new price series to verify its tradability and usefulness.

4 Further Measurements and Tests

We have shown how to test a price series for the property of stationarity using the ADF-test and also how to construct an artificial cointegrated pair using the Johansen test. Since in the financial world we are dealing with a tremendous amount of different price series' such as stocks, futures, options, and all kinds of other derivatives (e.g., currency pairs) there are numerous possibilities to form cointegrated pairs using the Johansen test. Even if we only take the seven most liquid currency pairs (USDJPY, USDCAD, USDCHF, EURUSD, GBPUSD, AUDUSD, NZSDUSD) and construct pairs consisting of three of those currencies we have $\frac{7*6*5}{1*2*3} = 35$ different possibilities of pair combinations. Obviously, we will reject those pairs for trading whose confidence for cointegration is below a certain percentile (e.g., 10pct) or sort them w.r.t. decreasing confidences but in practice it is often useful to calculate more quantitative measures such as the Hurst Exponent to evaluate the quality of a cointegrated pair.

4.1 Hurst Exponent

The idea behind the Hurst-Exponent (H) was already discussed in section 1. Generally speaking, the Hurst-Exponent tells us if our underlying series is a Geometric Brownian Motion ($H \approx 0.5$), mean reverting ($H < 0.5$), or trending ($H > 0.5$), where $0 < H < 1$. With H being smaller than 0.5 and close to 0 we can be more confident to have an underlying mean reverting price series.

There are several ways to approximate H . One simple approach can be seen in the following code snippet:

```
hurstApprox <- function(x){  
  tau <- vector()  
  lagvec <- vector()  
}
```

```

x <- as.numeric(x)

# Step through different lags
for(lag in seq(1,3,1)){
  # produce price difference with lags
  pp = x[(lag+1):length(x)] - x[1:(length(x)-(lag))]

  # write the different lags into a vector
  lagvec <- c(lagvec, lag)

  # calculate the variance of the difference vector
  tau <- c(tau, sqrt((sd(pp))))
}

# linear fit to double-log graph (gives power)
m <- polyfit(x=log10(lagvec), y=log10(tau), n=1)
hurst = m[1] * 2

return(hurst)
}

```

Unfortunately, our sample space is finite and, thus, the approximation might not be good at all. To verify our approximation of H we can however use another statistical test, namely, the Variance Ratio Test (VRT) explained in the following subsection.

4.2 Variance Ratio Test

To test for the statistical significance of our previously approximated Hurst Exponent we need a new statistical test to check whether our Null-hypothesis of $H = 0.5$ can be rejected or not. Therefore, (Lo and MacKinlay, 1988) developed a test statistics to address this issue (for more details see (Lo and MacKinlay, 2001)).

The idea behind this test is rather simple: It tests whether the equation

$$\frac{\text{Var}(z(t) - z(t - \tau))}{\tau \text{Var}(z(t) - z(t - 1))} = 1 \quad (3)$$

holds or not (or can be rejected or not respectively). In other words, if we hypothesize that our price series is indeed stationary than we expect that the variance of the

series will not increase over time, contrarily, if our series has a unit root and thus is trending or non-stationary we expect increasing variance over time. The idea is now to compare the variances of differently sampled subsets of our price series over time and check what happens with the ratio of the obtained variance (see eq. 3). Thus, the variance of the price series is calculated at Δt time periods. If we now sample every $k * \Delta t$ periods the variance is expected to be $k\sigma^2$ under a random walk, thus, equation 3 holds.

Summarizing, equation 3

- equals 1 if the price series follows a random walk,
- is (strictly) smaller than 1 under mean reversion (stationarity),
- and is (strictly) greater than 1 under mean aversion (trending up- or downwards).

Multiple values for k (usually $k = 2, 4, 8, 16, 32, \dots$) can be tested for different sampling periods allowing us to see at which intervals a price series may be trending and at which it may be mean reverting.

The Variance Ratio Test (VRT) is included in the R package *vrtest* (function *Lo.Mac*). Since it is too slow for the huge amount of data we have also implemented it in C++ with bindings to R.

4.3 Half Life of Mean Reversion

Now that we introduced tests and properties of a stationary price series one last helping feature will be discussed, namely, the Half Life of mean reversion.

The Half Life of the mean reversion is not directly used to measure or excess the quality of stationarity but will help us to develop a proper strategy to apply mean reversion within a trading framework. The Half Life time tells us how much time the price usually needs to revert to the series' mean.

The Half Life is accessed using the differential equation known as the *Ornstein-Uhlenbeck* formula (Uhlenbeck and Ornstein, 1930):

$$dy(t) = (\lambda y(t - 1) + \mu)dt + d\epsilon \quad (4)$$

with $d\epsilon$ being some Gaussian noise added to the process. As can be seen in the above single differential equation λ measures the "speed of the diffusion to the mean" or simply put the mean reversion speed. Though we can obtain the Half Life using equation 4 for us it is more appropriate to exploit some features we calculated beforehand, namely, the eigenvalues we obtained from the Johansen Test in subsection 3.2. To compute the approximate Half Life time of mean reversion we therefore use: $HL = \log(2)/Eigenvalue$.


```
summary(result)@lambda
## [1] 1.063e-02 6.912e-03 1.264e-03 -2.069e-18
# Half Life times for all eigenvalues
round(log(2)/result@lambda,2)
## [1] 6.520e+01 1.003e+02 5.482e+02 -3.350e+17
```

As it follows from above, our price series from subsection 3.2 has a Half Life time of mean reversion of about 65 periods (or 65h, since one period corresponds to one hour). Later we will use the Half Life time as an initial value for our lookback period optimization to compute moving averages and standard deviations to determine our trading entries and exits. Note: negative eigenvalues are invalid as they clearly state that no mean reversion exists.

Application on Foreign Exchange Markets

5 Introduction

In the previous chapter 1 we have introduced all important techniques and concepts of stationarity and cointegration. In the following section we will now continue to apply these concepts for a specific trading strategy used in the Foreign Exchange (Market) (FX). We will start to outline the general idea first and subsequently discuss important details. Additionally, we will provide a complete backtest using the major USD-pairs using hourly closing data from 2008-2013. Finally, we will end with some final notes about further considerations w.r.t. the presented strategy.

6 A Strategy for FX Markets

Though cointegration analysis and pair trading can be used on arbitrary markets we have chosen to focus exclusively on FX markets. This has rather practical reasons: First, the retail FX brokers provide customers with a tremendous amount of freely available data without any subscriptions fees or forced lags (e.g., when price data is lagged by a specific amount of time - often 10-15 minutes). Secondly, there are almost no technical and financial obstacles to trade the FX markets because of the vast amount of retail trading software and low capital requirements due to leverage.

6.1 Data, Data Processing, and Time Period

The underlying data used to evaluate the strategy was extracted from the Metatrader 4 software ¹ which was connected to an Alpari ² trading account. Note that Metatrader 4 was only used to have a historical and real time data feed and not for processing the data. The data was then forwarded using a slightly modified version of the freely available mt4r DLLs from Bernd Kreuss ³ into an R session for actual processing. Additionally, since the PIP-decimal point of different currency may differ the price data was normalized to PIPs (e.g., a price quote of 1.34128 EUR/USD was normalized to 13412.8 PIPs and a quote of 102.582 USD/JPY was normalized to 10258.2 PIPs).

We restricted ourselves to USD currency pairs to have only pairs with high liquidity and to make sure that risk measurements (in \$) are straightforward since the trading

¹ <http://www.metaquotes.net/en/metatrader4>

² <http://www.alpari.co.uk>

³ <https://sites.google.com/site/prof7bit/r-for-metatrader-4>

Fig. 3: A cointegrated pair with standard deviation (white horizontal lines) as potential entry signals.



account currency is assumed to be in U.S. Dollars (\$) as well. E.g., when a trade would be opened for a non-U.S.-\$ pair like AUD/CAD but the account base currency is in U.S.-\$ then we have an additional risk since we need to convert our transaction in AUD and CAD back to U.S.-\$. Thus, we restrict ourselves to U.S.-\$ pairs.

Furthermore, to reduce processing time and handle the complexity of the huge amount of data that could be used we have restricted ourselves to use one-hour closing times (1H closings) for this strategy though arbitrary time periods are possible if the hardware in use is sufficient.

In summary, the strategy explained below was developed for the 1H closings of the conversion rates of EUR/USD, AUD/USD, GBP/USD, NZD/USD, USD/CHF, USD/JPY, and USD/CAD.

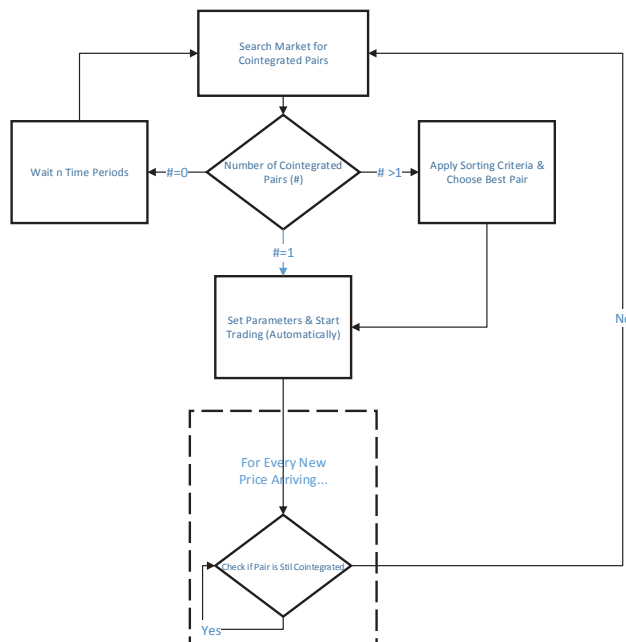
6.2 A Basic Strategy

The basic idea of the strategy is to find highly stationary pairs consisting of three cointegrated currency pairs (three pairs were chosen arbitrarily but with the benefit of hindsight it provides good results with acceptable monetary risk) and trade the mean reversion of the price of this cointegrated pair. Thus, we would enter a long position if the price is significantly *below* the mean price and enter a short position if the price is significantly *above* the mean (see Figure 3 for a simple example). In both cases we would exit our trade if the price has reverted, thus, the price hit the mean.

The above description of how to trade the cointegrated pair is rather intuitive but very vague. Therefore, we need to be more detailed in a few things, namely, we need to define what a good or highly stationary pair is and what quality criteria we can use to measure its "tradability". Finally, we need to be more precise of our trading rules, specifically, when to enter a trade.

The general process can be seen in the diagram below:

Fig. 4: The software's operating work flow.



we have already discussed the first step of scanning the market for cointegrated pairs and will now continue to explain which sorting and filter criteria were used to chose the most suitable pair for trading and when we have to stop trading and need to repeat the search for new cointegrated pairs. Finally, we will briefly present some evaluation results and conclude this report with a short discussion of what can be improved and what needs to be tested more extensively.

6.3 Sorting Criteria for Choosing a Pair to Trade

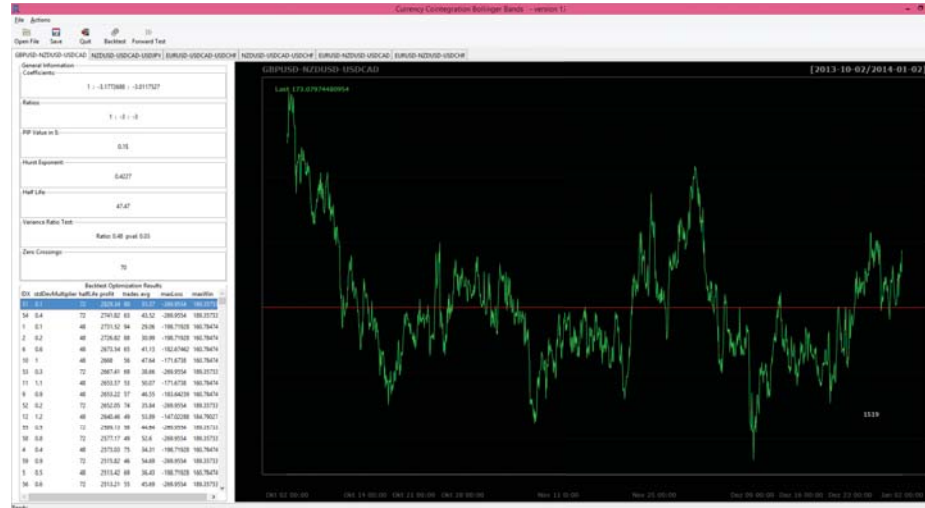
In chapter 1 we have already discussed several measures which can be directly used to sort all potential cointegrated pairs found for trading. Generally, we have to distinguish between two major ways when choosing which pair we want to use for trading: First, we can use intrinsic properties of the cointegrated price series such as the certainty of cointegration and stationarity which can be measured using the p-value of the ADF test (subsection 2.1), the variance ratio test result (subsection 4.2), or the Hurst-Exponent (subsection 4.1). Second, it is a good idea to use properties derived from backtesting the pair in question such as profits and losses, number of trades, balance/ equity draw downs, and average holding time for a position (note that we can use the Half Life time to get a good approximation about the expected holding time, see 4.3).

Though, we have not found a good general optimization scheme to directly chose the optimal pair for trading since we do not know which constraints and weights are best to set up an optimization task we found that it is best practice to first sort the found cointegrated pairs using intrinsic price series' characteristics (in this case the p-value of the ADF-test in increasing order - the lower the p-value the higher the certainty that we can reject the Null-hyphesis of a random walk) and subsequently scan the pairs w.r.t. trading properties such as profit & loss and average gain per trade. Figure 5 shows the implemented software after it found 6 cointegrated pairs and sorted them w.r.t increasing p-values.

6.4 Entries and Exits

Intuitively, we could calculate the standard deviation of the prices from a cointegrated pair and open a short trade if the price hits or crosses the standard deviation (betting that the price will go *down* to reach the mean price) or a long trade if the price hits or crosses the negative standard deviation (betting that the price will go *up* to reach the mean price). Here, we would exit our positions if the price hits the mean. However, there are two major problems that we encounter with this method: First, we do not know a-priori if the standard deviation is a good entry signal (note that the prices might not follow a Gaussian distribution) or if we should better use some multiple k of the standard

Fig. 5: Overview of the software after the search for cointegrated pairs was invoked. The found pairs are ordered by increasing p-value and backtesting results are presented in the lower left corner in tabular view.



deviation ($k * StandardDev$) for our entries. Secondly, though the process might remain stationary the actual mean of the price series might change slightly with increasing times, thus, a continuous updating of the mean might be appropriate to allow for small shifts of the time series' mean.

Hence, we have two parameters that need to be optimized in our lookback period before trading a pair: a) the so called z-score (the k used as a coefficient for the standard deviation) and b) the window length for computing the entry bound ($k * StandardDev$) and the mean of our price.

We need to optimize the two parameters to evaluate how the chosen pair performed if we would have traded it. Intuitively, we chose the Half Life of mean reversion as an initial value for our window length for computing a moving mean and moving standard deviation as any window length below the estimated Half Life is not expected to be reliable since it would be "faster" than the mean reversion itself. After choosing the initial window length to be the approximated Half Life time we furthermore test multiples of this window length to get optimal backtest trading results (in this case the software tests up to a multiple of 4 in steps of 0.5). For the second parameter, the z-score, we need to test different parameters for k ($k * StandardDev$) to find the optimal lower and upper moving bounds, since they are computed using a moving window with

a window length as described before) for our trading entries (those bands are commonly known as Bollinger Bands w.r.t. trading and technical analysis). Therefore, we start with $k = 0.1$ and test until $k = 5$ in steps of 0.1. Note that the upper trade entry bound for short trades is, thus, $k * StandardDev + MovingMean$ and the lower entry bound for long entries is $-k * StandardDev + MovingMean$.

As it can be seen in Figure 5 the software automatically performs backtests with the above described parameter combinations and presents the result in the lower left corner as a column wise sortable table. Moreover, the software allows us to get more detailed backtest result when a parameter combination is selected in the table and the backtest button is invoked (see Figure 6). This allows to include more information when choosing which pair will be traded and with what trading parameters. Please note that for all backtests and forward tests bid/ask spreads are always included into the results⁴.

With all the information above we then decide for one of the pairs to be traded and either trade it live using the Metatrader 4 Software or simulate a forward test using the implemented software. Although we have presented multiple ways of sorting the found cointegrated pairs it still has to be stressed that the decision of which pair to trade is highly subjective and has to be the topic of further research.

6.5 When to Stop Trading

Albeit the possibility of manually stopping the trading strategy (e.g., after a huge equity draw down or major news events) we need to check automatically when the cointegration of the pair we currently trade breaks, thus, if the p-value of the ADF-test exceeds a certain threshold (here we chose 0.2). Hence, if a new period is added (one hour passed) the software automatically reapplies the ADF-test on the cointegrated pair and if the p-value of the ADF-test exceeds 0.2 we close all position regardless if in profit or loss and stop trading. We then go back and try to find new cointegrated pairs and continue trading a new pair.

Notes about the Stop Loss Obviously, we can set an arbitrary stop loss helping us to exit a position if our equity draw down is too high for our current account balance. In common momentum (e.g., trend following) strategies this technique is useful and helps to protect traders against going broke in case of unlikely events such as significant news, flash crashes, or simply bad trading decisions. While it is useful to implement an emergency stop loss to prevent going broke a normal stop loss might be unwise

⁴ The overall s-max spread from <http://www.mt4i.com/spread/broker.aspx?brokerid=12> was chosen as a fixed spread for each corresponding pair.

Fig. 6: Example of the detail view for backtesting a selected pair with chosen parameters. The parameters for the backtest were chosen by selecting the corresponding row out of the table as shown in Figure 5. Green line: equity; red line: balance.



when trading mean reverting pairs. Assume, we have entered a long position 100PIPs below the mean of a cointegrated pair hoping that the price will go up to revert to its moving mean, however, the price does the opposite, thus, going down even more till it is 200PIPs below the mean. We have now a net loss of 100PIPs. We could have had avoided this situation if a simple Stop Loss of e.g. 50PIPs was used. However, our statistical assumption of mean reversion tells us that the price is more likely to mean revert the more away it is from its mean. Thus, when we would have exited after a loss of 50PIPs our basic strategy would have told us to re-enter the trade and still wait for mean reversion. So in the end it would have made no sense to exit the trade with a Stop Loss beforehand.

Because of the reasoning above, the only "Stop Loss" used during my backtest and forward tests was an emergency Stop Loss, that is, all position were closed when the trading pair was not stationary anymore *or* the current equity draw down while holding a position was higher than the highest equity draw down during the previously investigated backtests of the current pair (e.g., Figure 6 tells us that the maximum equity draw down was ≈ 345 PIPs and subsequently our emergency Stop Loss would have been set to 345 when trading this cointegrated pair).

6.6 Forward Test Results

The trading software we have implemented is capable to perform forward tests of cointegrated pairs using history data from Metatrader 4 as described before. Therefore, after a cointegrated pair and its parameters were selected the forward simulation is invoked and the trading is simulated until the pair is not cointegrated any more.

As a lookback period we have chosen a time span of two years. Hourly closing data of EUR/USD, GBP/USD, AUD/USD, USD/CAD, USD/JPY, NZD/USD, and USD/CHF were used to form the cointegrated pairs. Afterwards, a pair is chosen from the cointegrated series' found and was traded until cointegration broke. When this happened, the cointegration test was repeated from this hour on and a new pair was chosen to trade with. Though the lookback time can be chosen arbitrary it generally seems to hold that the longer the lookback is the more stable the cointegration seems to be - but the lookback itself might also be a optimization parameter for further research in this topic. Please note that the pairs were not cherry picked in any way but chosen after evaluating all the criteria discussed in 6.3 and 6.4.

The forward test results are displayed in Figure 7. The green line represents the equity curve and the red line the account balance (the initial balance was set to 0 so that a negative balance is possible). Furthermore, all returns, profits, losses, draw downs, etc. are measured in PIPs to be independent of ones individual capital and chosen leverage.

The overall performance of this rather simple strategy is astounding though a relatively high initial account balance is necessary to cope with rather large absolute draw downs. Though this forward test is made as realistic as possible we do not want to go into any more detail w.r.t. performance analytics despite the values shown below the equity/ balance chart in Figure 7 since this forward test was only made to illustrate the general capabilities of the trading strategy and its underlying software written in R.

7 Improvements and Further Considerations

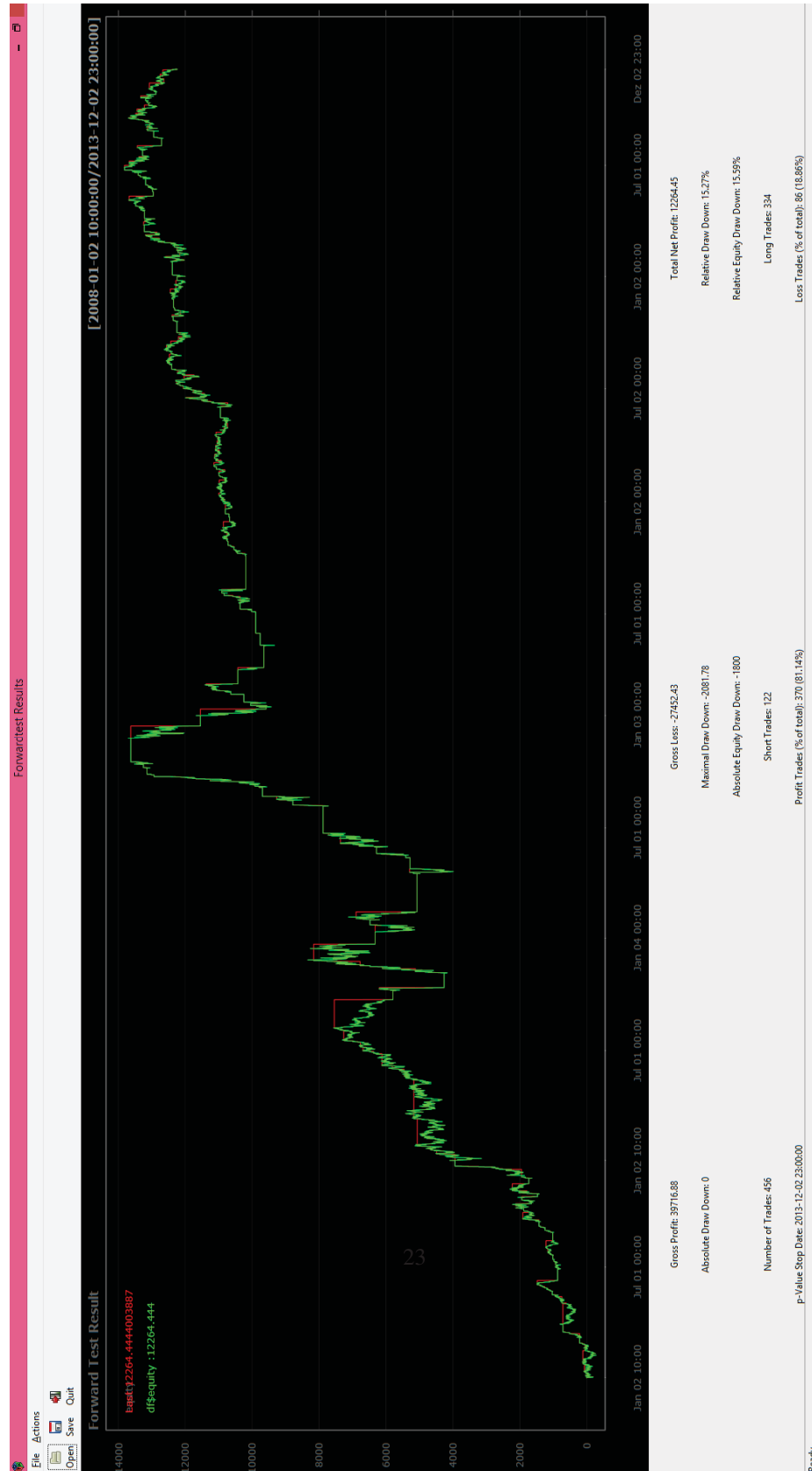
In 6.2 we have presented a basic strategy which applies the theoretical foundations of chapter 1 into a rather simple framework for trading on the Foreign Exchange Markets. Though the strategy performed well in the past it has to be stressed that there is a tremendous amount of work to fully incorporate a complete and reliable backtesting of such a trading system. Therefore, several questions have to be revisited and answered carefully for further applications of such trading systems:

- What time frame should be used for mean reversion trading?
- Which price information should be used to cointegrate the data (Open, High, Low, Close, or a combination of them)?

- What is a good time horizon for the lookback period? Could we exploit cointegration within a one year lookback or even one week?
- Are there any better signals of when to enter a trade despite using Bollinger Bands (e.g., Relative Strength Index (RSI))?
- Are there any better markets than the FX market to apply this strategy on?

To answer the questions raised above one has to extensively evaluate different parametrized and optimized instances of the basic strategy we have introduced. In summary, the basic idea behind cointegration has high potential to be further optimized and extended in almost every aspect. To do so the software was implemented platform independent and generic to allow testing of arbitrary data, time periods, number of pairs, and entry signals for further testing.

Fig. 7: Evaluation results for the simulated forward test from 2008 – 2013. Green line: equity; red line: balance.



References

- Chan, E. P. (2013). *Algorithmic Trading - Winning Strategies and Their Rationale*. John Wiley & Sons, Inc., Hoboken, New Jersey.
- Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a):427–431.
- Engle, R. F. and Granger, C. W. J. (1987). Co-integration and Error Correction: Representation, Estimation, and Testing. *Econometrica*, 55(2):251–76.
- Johansen, S. (1991). Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models. *Econometrica*, 59(6):1551–80.
- Lo, A. and MacKinlay, A. (1988). Stock market prices do not follow random walks: evidence from a simple specification test. *Review of Financial Studies*, 1(1):41–66.
- Lo, A. and MacKinlay, A. (2001). *A Non-Random Walk Down Wall Street*. Princeton University Press.
- Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Phys. Rev.*, 36:823–841.