Building the Money Machine

Or,

Examining the Efficient Markets Hypothesis in the spot cross currency foreign exchange markets

COMPUTER BASED FINANCIAL MODELLING
Professor Kiriakos Vlahos

Submitted by: Mukul Pareek, EMBA-Global 2003

CONTENTS

1. OVERVIEW

1.1. The foreign exchange markets

The market for foreign exchange is the world's largest financial market. Trading is conducted through a network of traders from major commercial and investment banks linked by phone, computer terminal and other telecommunications devices. Major trading centers include London, New York and Tokyo, with total trading volume in excess of a trillion dollars of foreign currency per day. Participants include importers and exporters, as well as traders, portfolio managers and foreign exchange brokers. This is an extremely liquid market that displays a high degree of market efficiency.

1.2. Spot trading

Spot trading in the foreign currency markets is for immediate delivery, though in reality it often means T+2, i.e. trades are settled 2 days after the transaction is done. The majority of foreign currency transactions involve the US Dollar. Spot rates between two currencies are normally quoted in the format Bid/Offer. For example, a quotation on dollar / mark of 1.5825-30 means that the person (or bank) making the quote is willing to sell marks for dollars at the rate of 1 dollar = 1.5825 marks, and is willing to buy marks for dollars at the rate of 1 dollar = 1.5830.

Trades are expressed in exchange rate pairs, where the first currency in the pair is the object of the trade, for example, 1 million dollar/mark at 1.5825 means US Dollar 1 million exchanged for Deutsche Marks 1,585,000.

1.3. Cross rates

Most currencies are expressed against dollars – which means there is always a buy-sell spread that would normally make it difficult to make money by routing transactions through dollars. However, there are some cross rates that trade directly. A cross rate is a rate between two non US Dollar currencies. For example, there may be trading between British Pounds and the Euro. Therefore, we have a rate for the Dollar-Euro, we have a rate for the Dollar-Pound, and we also have a rate for Pound-Euro.

1.4. Arbitrage – the opportunity for profit

The notion of arbitrage is simple. It means making money by exploiting price differences in the value of an asset in different markets. This project looks at the possibilities of making money in the foreign exchange markets by buying and selling foreign currencies in the spot market using different cross currency rates.

1.5. Triangular Arbitrage

To understand triangular arbitrage, let us take a hypothetical example where the exchange rate between the dollar and the British Pound is denoted by $S_{\$/£}$ (“S” for Spot, and the subscripts denoting the currencies involved), and the exchange rate between the dollar and the German Mark is denoted by $S_{\$/DM}$. In this situation, the exchange rate between the German Mark and the British Pound, $S_{DM/£}$ should be

$$S_{DM/£} = \frac{S_{\$/£}}{S_{\$/DM}} .$$

For example, if the exchange rate between the dollar and the British Pound is $1.50/£1 and the exchange rate between the dollar and the German Mark is $.75/DM1, the exchange rate between the German Mark and the British Pound is

$$S_{DM/£} = \frac{\$1.50/£1}{\$0.75/DM1} ,$$

$$= DM2/£1 .$$

If the exchange rate between the German Mark and the British Pound were either greater or less than DM2/£1, then a triangular arbitrage opportunity will be available. For example, suppose that the Mark/Pound exchange rate were DM2.1/£1. Then a trader with two German Marks would (1) exchange them for $1.50 (2 x $.75/DM1). The $1.5 would then (2) be exchanged for one British Pound, which would then (3) be used to purchase DM 2.1, which is greater than the number of German Marks that the trader started with.

In our model, we will use a similar logic to examine market imperfections where such money making opportunities may exist. We will consider the US Dollar as the base currency – though any other currency can be chosen as well.

2. APPROACH TO THE PROBLEM

Any market imperfections between currencies cannot exist for long – more so in the electronic market place where information travels instantaneously. Since the time window available for exploiting any such market imperfection will be very small, and the number of possible combinations of currency "round-trips" is very large, a computer necessarily needs to be employed to identify any profitable opportunities.

2.1. What is a "round-trip"?

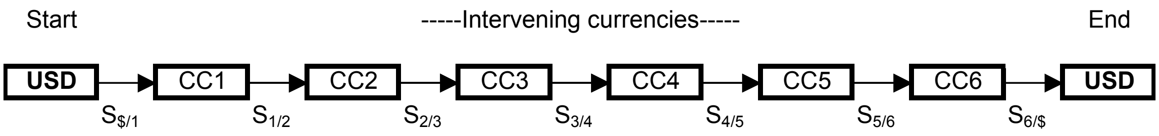
As explained under triangular arbitrage earlier, as different currency pairs are traded in different markets and the exchange rates may be momentarily out of sync, this may allow the arbitrageur to make a small profit by doing such a "round-trip", say for example by converting USD→GBP→EUR→USD. But there is no reason why a "round-trip" should be limited to two intervening currencies. It is possible to do much longer round-trips, for example USD→CHF→EUR→JPY→AUD→USD.

2.2. Constructing the "round-trips"

For any given number of currencies 'N', the number of cross-rates theoretically possible is N x (N-1)/2. The number of variations possible with N currencies to construct a "round-trip" would be N^N, which means that the number of possibilities increases exponentially as we increase the number of currencies being considered. The table below shows the number of combinations that we will need to consider for ‘N’ currencies.

Number of currencies	Number of "round trip" combinations possible
1	1
2	4
3	27
4	256
5	3125
6	46656
7	823543
8	16777216
9	387420489
10	10000000000

For example, consider a “round-trip” with 6 currencies – beginning and ending with USDs. This “round-trip” can be diagrammatically represented as follows:



$S_{i/j}$ represents the spot rate between currency i and j, and is used to convert from CCi to CCj

Note that each of the intervening currencies CC1 to 6 can be any of the 6 currencies, there is no limitation that a currency trade cannot be done twice, therefore the number of possible combinations is 6 x 6 x 6 x 6 x 6 x 6, or 6^6. This is important because it allows us to look at profitable combinations where the number of currencies is less than six. In other words, if our model suggests the following set of currency transactions:
USD → USD → USD → CHF → AUD → USD → USD → **USD**,
then it is the same as the set of transactions **USD** → CHF → AUD → **USD**.

This gives us complete flexibility in examining the entire range of possibilities that are available with the given set of currencies.

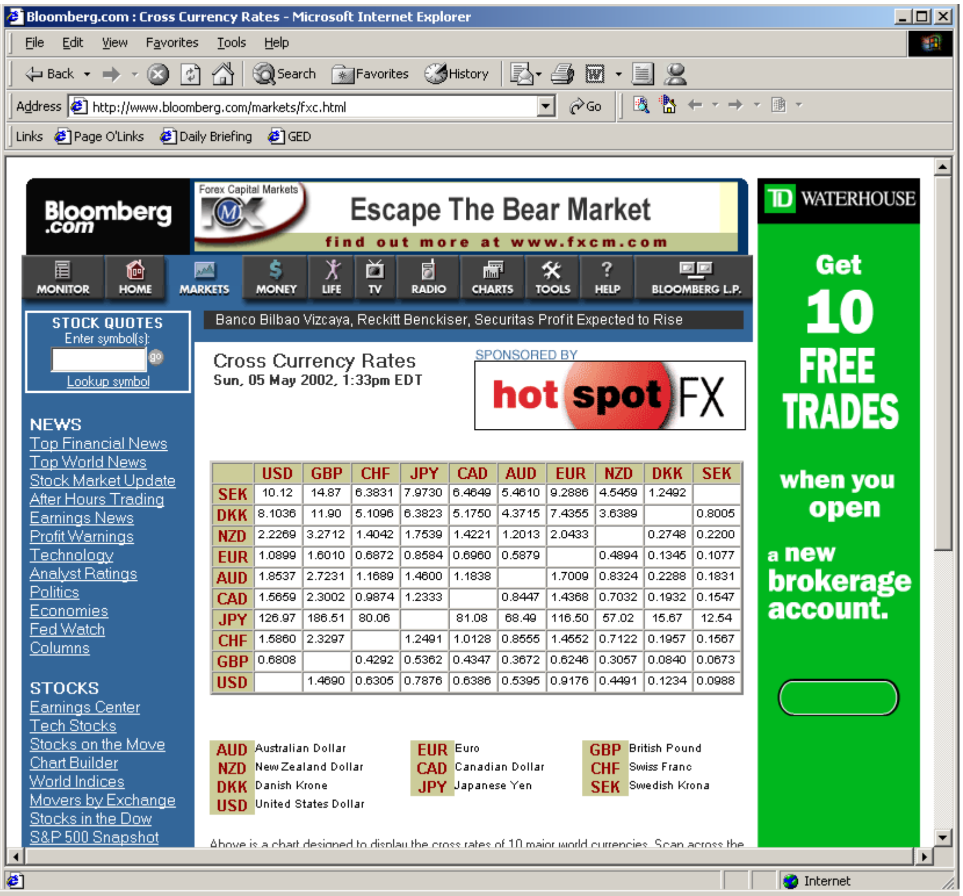
However, as the number of possible “round-trips” increases very rapidly as we increase the number of currencies, the model examines only “round-trips” with six currencies otherwise the computational time required goes up significantly without any additional contribution to our understanding of market efficiency.

The spreadsheet model built as part of this project examines the possibilities for such “round-trips” that can help make money by moving across currencies. It obtains the exchange rates from a trusted source on the web and examines all possible currency round-trips, and identifies the most profitable trading combination.

3. FORMULATING THE MODEL

3.1. Exchange rates

Bid and offer exchange rates are known, and an example of the rates is shown in the table below. In reality, bid-offer rates can be indicated in many different formats, however what is important is that each of the rates is known and can be plugged into a table with a format similar to the one below. The source of the data currently used in the model is http://www.bloomberg.com/markets/fxc.html, a screen shot of which is produced below.


Bloomberg.com : Cross Currency Rates - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.bloomberg.com/markets/fxc.html Go
Links Page O'Links Daily Briefing GED
Bloomberg.com
Forex Capital Markets
Escape The Bear Market
find out more at www.fxcm.com
MONITOR HOME MARKETS MONEY LIFE TV RADIO CHARTS TOOLS HELP BLOOMBERG L.P.
STOCK QUOTES
Enter symbol(s):
Lookup symbol
NEWS
Top Financial News
Top World News
Stock Market Update
After Hours Trading
Earnings News
Profit Warnings
Technology
Analyst Ratings
Politics
Economies
Fed Watch
Columns
STOCKS
Earnings Center
Tech Stocks
Stocks on the Move
Chart Builder
World Indices
Movers by Exchange
Stocks in the Dow
S&P 500 Snapshot
Banco Bilbao Vizcaya, Reckitt Benckiser, Securitas Profit Expected to Rise
SPONSORED BY
hot spot FX
TD WATERHOUSE
Get 10 FREE TRADES when you open a new brokerage account.
Internet

Cross Currency Rates
Sun, 05 May 2002, 1:33pm EDT

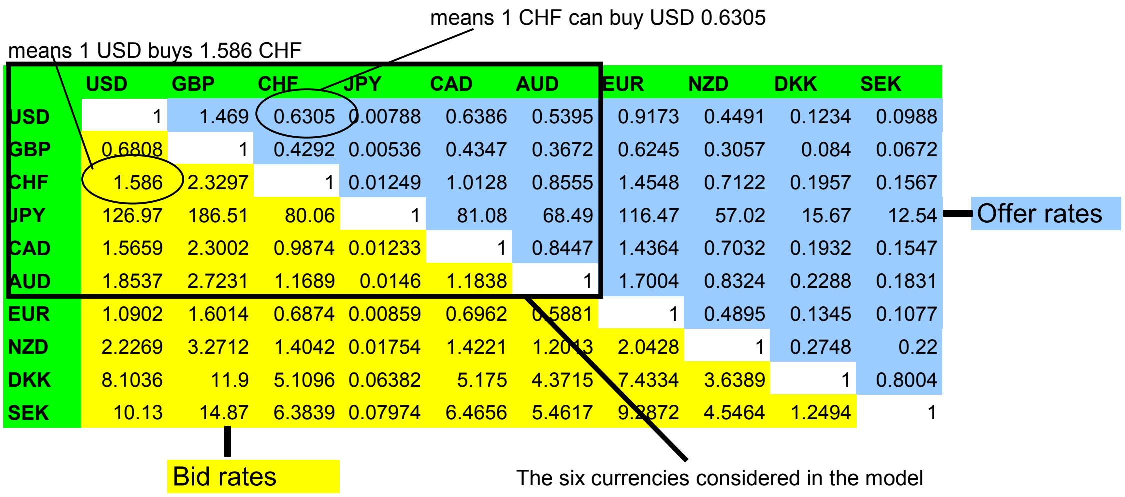
	USD	GBP	CHF	JPY	CAD	AUD	EUR	NZD	DKK	SEK
SEK	10.12	14.87	6.3831	7.9730	6.4649	5.4610	9.2886	4.5459	1.2492	
DKK	8.1036	11.90	5.1096	6.3823	5.1750	4.3715	7.4355	3.6389		0.8005
NZD	2.2269	3.2712	1.4042	1.7539	1.4221	1.2013	2.0433		0.2748	0.2200
EUR	1.0899	1.6010	0.6872	0.8584	0.6960	0.5879		0.4894	0.1345	0.1077
AUD	1.8537	2.7231	1.1689	1.4600	1.1838		1.7009	0.8324	0.2288	0.1831
CAD	1.5659	2.3002	0.9874	1.2333		0.8447	1.4368	0.7032	0.1932	0.1547
JPY	126.97	186.51	80.06		81.08	68.49	116.50	57.02	15.67	12.54
CHF	1.5860	2.3297		1.2491	1.0128	0.8555	1.4552	0.7122	0.1957	0.1567
GBP	0.6808		0.4292	0.5362	0.4347	0.3672	0.6246	0.3057	0.0840	0.0673
USD		1.4690	0.6305	0.7876	0.6386	0.5395	0.9176	0.4491	0.1234	0.0988

AUD	Australian Dollar	EUR	Euro	GBP	British Pound
NZD	New Zealand Dollar	CAD	Canadian Dollar	CHF	Swiss Franc
DKK	Danish Krone	JPY	Japanese Yen	SEK	Swedish Krona
USD	United States Dollar				

Above is a chart designed to display the cross rates of 10 major world currencies. Scan across the

Reading the exchange rates matrix

The exchange rates from the web need to be "sanitised" and properly formatted and arranged as to facilitate their use in VBA arrays and macros. The final rearranged set of exchange rates – which is not too different from the raw data from the web query is explained below:



means 1 CHF can buy USD 0.6305

means 1 USD buys 1.586 CHF

	USD	GBP	CHF	JPY	CAD	AUD	EUR	NZD	DKK	SEK
USD	1	1.469	0.6305	0.00788	0.6386	0.5395	0.9173	0.4491	0.1234	0.0988
GBP	0.6808	1	0.4292	0.00536	0.4347	0.3672	0.6245	0.3057	0.084	0.0672
CHF	1.586	2.3297	1	0.01249	1.0128	0.8555	1.4548	0.7122	0.1957	0.1567
JPY	126.97	186.51	80.06	1	81.08	68.49	116.47	57.02	15.67	12.54
CAD	1.5659	2.3002	0.9874	0.01233	1	0.8447	1.4364	0.7032	0.1932	0.1547
AUD	1.8537	2.7231	1.1689	0.0146	1.1838	1	1.7004	0.8324	0.2288	0.1831
EUR	1.0902	1.6014	0.6874	0.00859	0.6962	0.5881	1	0.4895	0.1345	0.1077
NZD	2.2269	3.2712	1.4042	0.01754	1.4221	1.2013	2.0428	1	0.2748	0.22
DKK	8.1036	11.9	5.1096	0.06382	5.175	4.3715	7.4334	3.6389	1	0.8004
SEK	10.13	14.87	6.3839	0.07974	6.4656	5.4617	9.2872	4.5464	1.2494	1

Offer rates

Bid rates

The six currencies considered in the model

3.2. Considerations

Investment

The model considers all possible combinations of currencies in a 6-currency "round-trip" (surrounded by USD on each side), and identifies the profit maximising "round-trip" for an investor who "invests" USD 1 million into the round-trip.

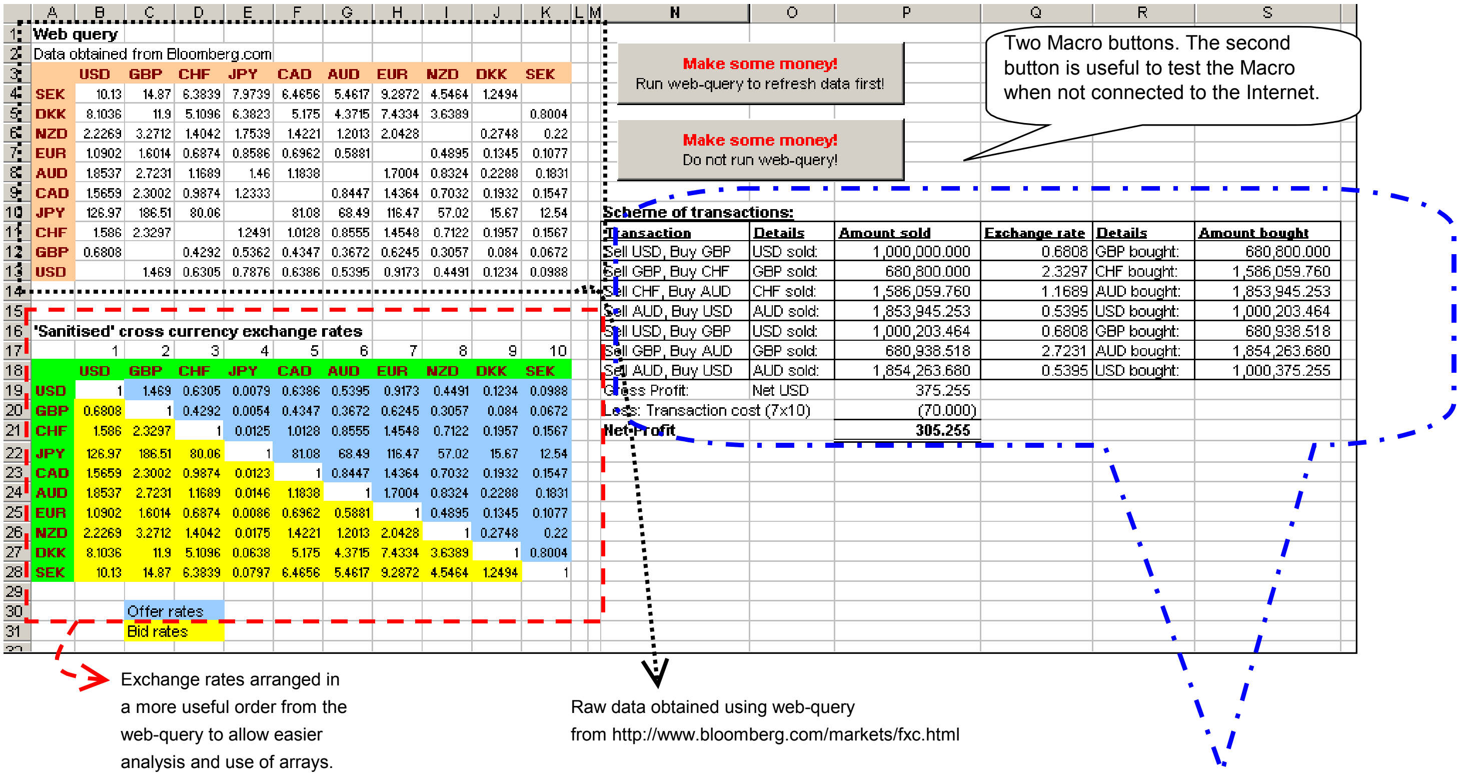
It is important to note that under spot transactions this "investment" is recouped instantaneously, and is therefore not really an investment but just a convenient number. The amount of USD 1 million is entirely arbitrary, we can use any amount, even $1, only that the profits will be proportionately less. The amount of $1 million was chosen to see round numbers that are easier to understand than a large number of decimals.

Considering bid-offer spreads: Bid offer spreads (that are really a form of transaction costs) are considered as part of the exchange rates table mentioned earlier.

Considering transaction costs: Large institutional investors do not normally pay per-transaction costs of trading in the foreign exchange markets. However, individuals or smaller investors normally incur a fixed fee per transaction. This fixed fee is user specified as an input parameter when the model is run. It can be specified as "0" (zero) if that be the case.

3.3. Understanding the model

The following is a screen shot of what the model looks like, together with an explanation of each section of the spreadsheet.



Web query

Data obtained from Bloomberg.com

	USD	GBP	CHF	JPY	CAD	AUD	EUR	NZD	DKK	SEK
SEK	10.13	14.87	6.3839	7.9739	6.4656	5.4617	9.2872	4.5464	1.2494	
DKK	8.1036	11.9	5.1096	6.3823	5.175	4.3715	7.4334	3.6389		0.8004
NZD	2.2269	3.2712	1.4042	1.7539	1.4221	1.2013	2.0428		0.2748	0.22
EUR	1.0902	1.6014	0.6874	0.8586	0.6962	0.5881		0.4895	0.1345	0.1077
AUD	1.8537	2.7231	1.1689	1.46	1.1838		1.7004	0.8324	0.2288	0.1831
CAD	1.5659	2.3002	0.9874	1.2333		0.8447	1.4364	0.7032	0.1932	0.1547
JPY	126.97	186.51	80.06		81.08	68.49	116.47	57.02	15.67	12.54
CHF	1.586	2.3297		1.2491	1.0128	0.8555	1.4548	0.7122	0.1957	0.1567
GBP	0.6808		0.4292	0.5362	0.4347	0.3672	0.6245	0.3057	0.084	0.0672
USD		1.469	0.6305	0.7876	0.6386	0.5395	0.9173	0.4491	0.1234	0.0988

'Sanitised' cross currency exchange rates

	1	2	3	4	5	6	7	8	9	10
	USD	GBP	CHF	JPY	CAD	AUD	EUR	NZD	DKK	SEK
USD	1	1.469	0.6305	0.0079	0.6386	0.5395	0.9173	0.4491	0.1234	0.0988
GBP	0.6808	1	0.4292	0.0054	0.4347	0.3672	0.6245	0.3057	0.084	0.0672
CHF	1.586	2.3297	1	0.0125	1.0128	0.8555	1.4548	0.7122	0.1957	0.1567
JPY	126.97	186.51	80.06	1	81.08	68.49	116.47	57.02	15.67	12.54
CAD	1.5659	2.3002	0.9874	0.0123	1	0.8447	1.4364	0.7032	0.1932	0.1547
AUD	1.8537	2.7231	1.1689	0.0146	1.1838	1	1.7004	0.8324	0.2288	0.1831
EUR	1.0902	1.6014	0.6874	0.0086	0.6962	0.5881	1	0.4895	0.1345	0.1077
NZD	2.2269	3.2712	1.4042	0.0175	1.4221	1.2013	2.0428	1	0.2748	0.22
DKK	8.1036	11.9	5.1096	0.0638	5.175	4.3715	7.4334	3.6389	1	0.8004
SEK	10.13	14.87	6.3839	0.0797	6.4656	5.4617	9.2872	4.5464	1.2494	1

🟦 Offer rates

🟨 Bid rates

Scheme of transactions:

Transaction	Details	Amount sold	Exchange rate	Details	Amount bought
Sell USD, Buy GBP	USD sold:	1,000,000.000	0.6808	GBP bought:	680,800.000
Sell GBP, Buy CHF	GBP sold:	680,800.000	2.3297	CHF bought:	1,586,059.760
Sell CHF, Buy AUD	CHF sold:	1,586,059.760	1.1689	AUD bought:	1,853,945.253
Sell AUD, Buy USD	AUD sold:	1,853,945.253	0.5395	USD bought:	1,000,203.464
Sell USD, Buy GBP	USD sold:	1,000,203.464	0.6808	GBP bought:	680,938.518
Sell GBP, Buy AUD	GBP sold:	680,938.518	2.7231	AUD bought:	1,854,263.680
Sell AUD, Buy USD	AUD sold:	1,854,263.680	0.5395	USD bought:	1,000,375.255
Gross Profit:	Net USD	375.255			
Less: Transaction cost (7x10)		(70.000)			
Net Profit		**305.255**			

Two Macro buttons. The second button is useful to test the Macro when not connected to the Internet.

Exchange rates arranged in a more useful order from the web-query to allow easier analysis and use of arrays. The JPY exchange rates have also been expressed correctly, and cross rates between the same currency have been expressed as "1" to allow lookups.

Raw data obtained using web-query from http://www.bloomberg.com/markets/fxc.html

The results. This shows the scheme of transactions to be entered into, and also all the related calculations that can be verified.

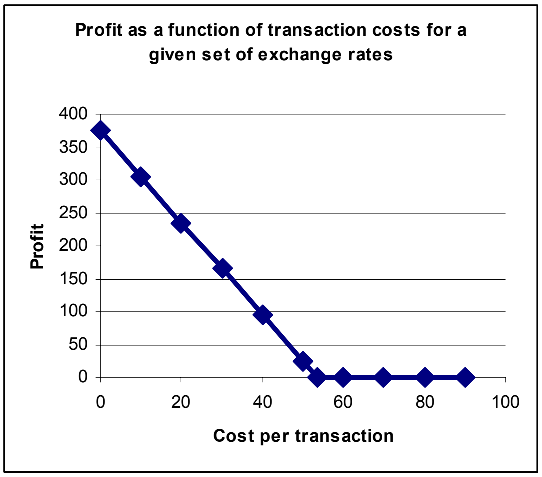
3.4. Using the model

Using the model is simple – by clicking either of the two buttons on the spreadsheet, we start the macro that analyses all the possibilities and lists the "round-trip" that gives the most profit. It also lists the scheme of transactions on the Excel Sheet together with all calculations that are easy to verify.

If there are no profitable possibilities, then the model says so, and recommends no transactions (i.e. all currencies are USD). (This branch of the model can be verified by the user specifying a very high cost per transaction.)

3.5. Sensitivity to transaction costs

The graph below shows total profit as a function of transaction costs for a given set of exchange rates. As expected, the graph is linear till transaction costs reach $53.61 after which there is no profit and the model will not recommend transacting any currencies.



3.6. Why was Excel's Solver not used?

Attempts were made to solve the problem in many different ways as to allow the use of Excel's Solver. This included a very frustrating attempt to model this as an "assignment problem" where currency inflows and outflows were constrained and the net USD flow was to be maximised. However, for some reason Excel's solver could not optimise the problem and would either leave the initial solution unchanged without optimising or return an error for no apparent reason. This was perhaps a result of discontinuities introduced by vlookups/matches on the exchange rates matrix. In the end, a brute force approach was quite productively used with Visual Basic cycling through each combination. It takes VB about 10 seconds to go through 46,000 combinations on a P3-500 MHz computer, which should be acceptable.

3.7. Model logic- how does it work?

The coding logic inherent in the model is quite straight forward and is summarised below:

1. Refresh the exchange rates data by running a web-query, and assign the resultant matrix to an array variable,
2. Ask the user for the transaction cost per foreign exchange transaction,
3. Use six nested "For…Next" loops to cycle through each possible combination of currencies,
4. Each time, calculate the profit that would result from executing the given combination of currency transactions, and if this profit is greater than the profit from a previous combination, then store it away in a variable, otherwise discard it. (The initial value of profit is 0)
5. If a profitable combination is not found – which for instance will be the case where transaction costs are very high, the macro says so.
6. If a profitable combination is found, the same is listed together with all combinations.

4. CRITIQUE

4.1. Is there really any money to be made from this model?

The model seems to suggest that there is some money to be made – even if it is a few hundred dollars on a USD 1 million transaction. Which is actually quite high given that the initial "investment" is recouped almost instantaneously with a certain profit and no risk.

Foreign exchange markets move very extraordinarily fast. In the real trading world rates move up and down many times each second and tend to remain in sync with each other. The data which this model is based upon is updated very infrequently by Bloomberg (once every 30 minutes during the daytime when the markets are open) and there is no guarantee as to timeliness or accuracy. Also, these rates are merely indicative and are not firm bid or offer rates.

The model can be truly tested only if it is based upon professional data – for example if sourced from live feeds Reuters PowerPlusPro which is a paid service allowing Excel integration of Reuters data feeds. It is very likely that such an exercise will reveal no profit opportunities, vindicating the efficient markets hypothesis.

4.2. So where are the opportunities?

Opportunities for extraordinary profits exist only in inefficient markets. Therefore, if any money has to be made, the model needs to be expanded to markets that are shallow or inefficient.

In economies where foreign exchange markets are regulated, there often exists a black market exchange rate that is different from the official rate. Arbitrage opportunities between these rates are always possible and there is some money to be made. However, this may involve some risk taking in the form of possible violations of foreign exchange regulations. The cost of these violations with the probability of being discovered will need to be factored into the model.

Also, markets for "exotic currency pairs" tend to be shallow and may offer brief opportunities for legitimate money to be made – for example the Thai Baht-Indian Rupee cross currency rates.

4.3. Enhancing the model

Better data

Using better data, for example from Reuters as suggested earlier, we can enhance the usefulness of the model.

Moving across time - beyond spot transaction trading

The model can be extended beyond spot transactions to look at forward transactions, interest rates, and identify any opportunities that may arise from an imbalance between the spot and forward rates given the interest rates in the two currencies. Such arbitrage would be based on violations of interest rate parity.

Better output
The output from the model is not very useful if it is listed as a series of transactions to be executed. Foreign exchange quotes are not valid for more than a few seconds and the opportunity may vanish by the time these deals are entered into. In order to be truly useful, a model such as this needs to be integrated into the dealing system whereby orders are placed instantaneously without any human intervention.

5. CONCLUSION

The modelling exercise brings to us the benefit of actually observing market efficiency in action. In spite of the very large number of cross-currency rates possible, there is little imbalance and all rates tend to move together. This is partly due to the fact that there are arbitrageurs across the world constantly monitoring exchange rates and quickly moving in for a free lunch as soon as the slightest opportunity presents itself.

REFERENCES:

1. *Managing Foreign Exchange Risk, David F. Derosa 1996*
2. *Arbitrage Pricing Theory by A. Penati & G. Pennacchi*
3. *Exchange Rates and Investment Decisions by Prof Day, University of Texas*

APPENDIX 1 - VB CODE USED

```
Option Base 1
'Option Explicit not used as it unnecessarily limits the use of variables 'on-the-go'
'while coding.
________________________________

Sub Macro2()
Range("fxc_2").Select
Selection.QueryTable.Refresh BackgroundQuery:=False
Macro1
End Sub
_________________________________

Sub Macro1()

'This section initialises the variables
    rates = Range("rates")      'the exchange rates
    ccy = Range("ccy")          'the currency codes
    Profit = 0                   'initial value for profit
    StartTime = Timer            'for timing the macro
    Application.ScreenUpdating = False

'Ask for user input for transaction costs
    UnitTxCost = InputBox("Enter transaction cost per transaction in USDs (e.g. 20)" & Chr(13) _
                & "Do not enter any text or the $ sign!", "Transaction Costs")

    If IsNumeric(UnitTxCost) = False Then UnitTxCost = 0

'Start the 6 nested loops to go through various currency combinations
    For c1 = 1 To 6
    For c2 = 1 To 6
    For c3 = 1 To 6
    For c4 = 1 To 6
    For c5 = 1 To 6
    For c6 = 1 To 6

'Update the status bar to show the combination being checked.
    Application.StatusBar = "Checking iteration " & ccy(1, c1) & "-" & ccy(1, c2) _
            & "-" & ccy(1, c3) & "-" & ccy(1, c4) & "-" & ccy(1, c5) & "-" & ccy(1, c6)


'Calculate true number of transactions. When the same currency is repeated, then it is
'not a transaction, for example USD->GBP->GBP->JPY->USD = USD->GBP->JPY->USD, ie 4 transactions
'and not 5, as GBP->GBP is costless. This is needed to calculate true transaction costs.

    TxNumber = 7
    If 1 = c1 Then TxNumber = TxNumber - 1
    If c1 = c2 Then TxNumber = TxNumber - 1
    If c2 = c3 Then TxNumber = TxNumber - 1
    If c3 = c4 Then TxNumber = TxNumber - 1
    If c4 = c5 Then TxNumber = TxNumber - 1
    If c5 = c6 Then TxNumber = TxNumber - 1
    If c6 = 1 Then TxNumber = TxNumber - 1

'For the currency combination in the loop, compare profit to previous highest profits.
'If the current combination yields higher profits, then "GoSub" to 'finito' where
'the currrent combination is stored away (to be replaced by something better in future
'iterations of the loop.

If Profit < ((1000000 * rates(c1, 1) * rates(c2, c1) * rates(c3, c2) * rates(c4, c3) _
            * rates(c5, c4) * rates(c6, c5) * rates(1, c6) - 1000000) - (TxNumber * UnitTxCost)) _
            Then GoSub finito


howmany = howmany + 1   'merely a count of how many combinations checked.


Next c6         'Increment and loop again
Next c5
Next c4
Next c3
Next c2
Next c1

'By now, the most profitable opportunity has been identified, and needs to be published
'on the Excel sheet and on the message box. (If there is no profitable opportunity, then
'the "GoSub finito" would not have been visited and x would be zero)
```

```
If x = 0 Then
    MsgBox ("No profitable possibilities!")

Else

    tc0 = ccy(1, 1)         'Replace currency numbers by their respective English
    tc1 = ccy(1, fc1)       'codes, eg 1 means USD etc
    tc2 = ccy(1, fc2)
    tc3 = ccy(1, fc3)
    tc4 = ccy(1, fc4)
    tc5 = ccy(1, fc5)
    tc6 = ccy(1, fc6)
    tc7 = ccy(1, 1)


    For n = 1 To 7  'Remove duplicates, ie "GBP->GBP" is merely "GBP" in the transaction.
        If tc0 = tc1 Then tc0 = ""
        If tc1 = tc2 Then tc1 = ""
        If tc2 = tc3 Then tc2 = ""
        If tc3 = tc4 Then tc3 = ""
        If tc4 = tc5 Then tc4 = ""
        If tc5 = tc6 Then tc5 = ""
        If tc6 = tc7 Then tc6 = ""
    Next n

End If

FinishTime = Timer      'End of processing

'Show message box with currencies to trade, profit, and the time taken and iterations checked.
MsgBox ("Transact currencies as follows:" & Chr(13) & _
        tc0 & "-" & tc1 & "-" & tc2 & "-" & tc3 & "-" & tc4 & "-" & tc5 & "-" & tc6 & _
        "-" & tc7 & Chr(13) & "Profit on a $1mio investment: $" & _
        Application.Text(Profit, "0.00") & Chr(13) & Chr(13) & _
        "(" & howmany & " possibilities checked in " & _
        FormatNumber(FinishTime - StartTime, 1) & " seconds.)")


'FROM THIS POINT ONWARDS, THE EXCEL SHEET RANGE N11:S21, 'SCHEME OF TRANSACTIONS', IS WRITTEN

    'Clear range and create headings
    Range("N11:S21").ClearContents
    Range("N11") = "Transaction"
    Range("O11") = "Details"
    Range("P11") = "Amount sold"
    Range("Q11") = "Exchange rate"
    Range("R11") = "Details"
    Range("S11") = "Amount bought"

    'Replace blanks with USDs.  All blanks will be in the beginning.
    If tc0 = "" Then tc0 = "USD"
    If tc1 = "" Then tc1 = "USD"
    If tc2 = "" Then tc2 = "USD"
    If tc3 = "" Then tc3 = "USD"
    If tc4 = "" Then tc4 = "USD"
    If tc5 = "" Then tc5 = "USD"
    If tc6 = "" Then tc6 = "USD"
    If tc7 = "" Then tc7 = "USD"

    'Write the transaction
    Range("N12") = "Sell " & tc0 & ", Buy " & tc1
    Range("N13") = "Sell " & tc1 & ", Buy " & tc2
    Range("N14") = "Sell " & tc2 & ", Buy " & tc3
    Range("N15") = "Sell " & tc3 & ", Buy " & tc4
    Range("N16") = "Sell " & tc4 & ", Buy " & tc5
    Range("N17") = "Sell " & tc5 & ", Buy " & tc6
    Range("N18") = "Sell " & tc6 & ", Buy " & tc7

    'Write the transaction details for currency sold
    Range("O12") = tc0 & " sold:"
    Range("O13") = tc1 & " sold:"
    Range("O14") = tc2 & " sold:"
    Range("O15") = tc3 & " sold:"
    Range("O16") = tc4 & " sold:"
    Range("O17") = tc5 & " sold:"
    Range("O18") = tc6 & " sold:"

    'Write the transaction details for currency bought
    Range("R12") = tc1 & " bought:"
    Range("R13") = tc2 & " bought:"
    Range("R14") = tc3 & " bought:"
    Range("R15") = tc4 & " bought:"
    Range("R16") = tc5 & " bought:"
    Range("R17") = tc6 & " bought:"
    Range("R18") = tc7 & " bought:"


    'Write the amounts
    Range("P12") = 1000000
    Range("Q12") = "=VLOOKUP(LEFT(RC[1],3),rates2,1+MATCH(LEFT(RC[-2],3),ccy,0),FALSE)"
    Range("S12") = "=RC[-2]*RC[-3]"
    Range("S13") = "=RC[-2]*RC[-3]"
    Range("S14") = "=RC[-2]*RC[-3]"
    Range("S15") = "=RC[-2]*RC[-3]"
    Range("S16") = "=RC[-2]*RC[-3]"
    Range("S17") = "=RC[-2]*RC[-3]"
    Range("S18") = "=RC[-2]*RC[-3]"
    Range("Q13") = _
        "=VLOOKUP(LEFT(RC[1],3),rates2,1+MATCH(LEFT(RC[-2],3),ccy,0),FALSE)"
    Range("Q14") = _
        "=VLOOKUP(LEFT(RC[1],3),rates2,1+MATCH(LEFT(RC[-2],3),ccy,0),FALSE)"
    Range("Q15") = _
        "=VLOOKUP(LEFT(RC[1],3),rates2,1+MATCH(LEFT(RC[-2],3),ccy,0),FALSE)"
    Range("Q16") = _
        "=VLOOKUP(LEFT(RC[1],3),rates2,1+MATCH(LEFT(RC[-2],3),ccy,0),FALSE)"
    Range("Q17") = _
        "=VLOOKUP(LEFT(RC[1],3),rates2,1+MATCH(LEFT(RC[-2],3),ccy,0),FALSE)"
    Range("Q18") = _
        "=VLOOKUP(LEFT(RC[1],3),rates2,1+MATCH(LEFT(RC[-2],3),ccy,0),FALSE)"
    Range("P13") = "=R[-1]C[3]"
    Range("P14") = "=R[-1]C[3]"
    Range("P15") = "=R[-1]C[3]"
    Range("P16") = "=R[-1]C[3]"
    Range("P17") = "=R[-1]C[3]"
    Range("P18") = "=R[-1]C[3]"
    Range("P19") = "=R[-1]C[3]-R[-7]C"
    Range("N19") = "'Gross Profit:"
    Range("O19") = "'Net USD"
    Range("N20") = "'Less: Transaction Cost (" & Application.Text(TxNumberFinal, "general") & _
                    " x $" & (Application.Text(UnitTxCost, "general")) & ")"
    Range("P20") = -UnitTxCost * TxNumberFinal
    Range("N21") = "'Net Profit"
    Range("P21") = "=R[-2]C+R[-1]C"

'END OF CODE FOR WRITING IN THE EXCEL SPREADSHEET

'Tidying up
'Clear the status bar
Application.StatusBar = ""

'Exit sub ahead of finito so finito does not get executed again unnecessarily in the end
Exit Sub

'Subroutine called finito - this could alternatively have been handled by calling another
'macro, but this works just as well

finito:
    fc0 = 1
    fc1 = c1
    fc2 = c2
    fc3 = c3
    fc4 = c4
    fc5 = c5
    fc6 = c6
    fc7 = 1
    x = 1
    Profit = ((1000000 * rates(c1, 1) * rates(c2, c1) * rates(c3, c2) * rates(c4, c3) * _
        rates(c5, c4) * rates(c6, c5) * rates(1, c6) - 1000000) - (TxNumber * UnitTxCost))
    TxNumberFinal = TxNumber
Return



End Sub
```