Lecture 5

- Foreign-Currency Trading
- Integer Programming
  - Plant-location example
- Summary and Preparation for next class
Foreign Exchange (FX) Markets

- FX markets are big
  - Daily trading often exceeds $1 trillion
  - Worldwide interbank market
- Many types of markets and instruments:
  - Spot currency markets
  - Forward and futures markets
- Derivative FX instruments include:
  - Currency options
  - Currency swaps

Sample of Uses of FX Instruments

- Corporations
  - Manage currency positions for international operations
  - Manage corporate currency risk
- Global-Investment Portfolios
  - Speculate in foreign-currency markets
  - Hedge currency risk in international equity investments
  - Hedge/speculate in global fixed-income markets
Foreign-Currency Trading

A **spot currency transaction** is an agreement to buy some amount of one currency using another currency.

- **Example 1:** At today’s rates, 10,000 U.S. dollars can be converted into 6,390 British pounds:

  \[
  10,000 \text{ US$} \times 0.6390 \text{ Pound/$} \rightarrow 6,390 \text{ British Pounds}
  \]

- **Example 2:** At today’s rates, 10,000 German D-Marks can be converted into 629,400 Japanese yen:

  \[
  10,000 \text{ DM} \times 62.94 \text{ Yen/DM} \rightarrow 629,400 \text{ Yen}
  \]

**Figure 1.** Today’s Cross-Currency Spot Rates

<table>
<thead>
<tr>
<th>To:</th>
<th>US Dollar</th>
<th>Pound</th>
<th>FFranc</th>
<th>D-Mark</th>
<th>Yen</th>
</tr>
</thead>
<tbody>
<tr>
<td>From: US Dollar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pound</td>
<td>0.6390</td>
<td>5.3712</td>
<td>1.5712</td>
<td>98.8901</td>
<td></td>
</tr>
<tr>
<td>FFranc</td>
<td>1.5648</td>
<td>8.4304</td>
<td>2.4590</td>
<td>154.7733</td>
<td></td>
</tr>
<tr>
<td>D-Mark</td>
<td>0.1856</td>
<td>0.1186</td>
<td>0.2921</td>
<td>18.4122</td>
<td></td>
</tr>
<tr>
<td>Yen</td>
<td>0.6361</td>
<td>0.4063</td>
<td>3.4233</td>
<td>62.9400</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To:</th>
<th>US Dollar</th>
<th>Pound</th>
<th>FFranc</th>
<th>D-Mark</th>
<th>Yen</th>
</tr>
</thead>
<tbody>
<tr>
<td>From: US Dollar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pound</td>
<td>1.5648</td>
<td>8.4304</td>
<td>2.4590</td>
<td>154.7733</td>
<td></td>
</tr>
<tr>
<td>FFranc</td>
<td>0.1856</td>
<td>0.1186</td>
<td>0.2921</td>
<td>18.4122</td>
<td></td>
</tr>
<tr>
<td>D-Mark</td>
<td>0.6361</td>
<td>0.4063</td>
<td>3.4233</td>
<td>62.9400</td>
<td></td>
</tr>
<tr>
<td>Yen</td>
<td>0.01011</td>
<td>0.00645</td>
<td>0.05431</td>
<td>0.01588</td>
<td></td>
</tr>
</tbody>
</table>
Transactions Costs

- For large transactions in the world interbank market, there are no commission charges. However, transactions costs are implicit in the bid-offer spreads.

- *Example 1 (cont’d):* At today’s rates, 6,390 British pounds can be converted into 9,999.07 U.S. dollars:

\[
6,390 \text{ British Pounds} \xrightarrow{1.5648 \$/\text{Pound}} 9,999.07 \text{ US$}
\]

Aside: Quotations are usually given as:

$ /\text{pound}: 1.5648-1.5649.

The rate 1.5648 is the *bid price* for pounds, i.e., it means that a bank is willing to *buy* a pound for 1.5648 dollars. The rate 1.5649 (= 1/0.6390) is the *offer price* for pounds, i.e., it means that a bank is offering to *sell* a pound for 1.5649 dollars. The bid-offer spread represents a source of profit for the market maker and a transaction cost for the counterparty in the transaction.
Arbitrage

- **Definition:** Arbitrage is a set of spot currency transactions that creates positive wealth but does not require any funds to initiate, i.e., it is a “money pump.”

- **Example:** Suppose that today’s pound/$ rate is 0.6390 and today’s $/pound rate is 1.5651. Then an investor could make arbitrage profits as follows:

  1 US Dollar → 6,390 pounds → 10,000.99 doll
  2 British pound → 6,390 pounds → 10,000

  6,390 pounds times 1.5651 $/pound = $10,000.99.

  These two transactions make $0.99 in arbitrage profit and require no initial investment.
Arbitrage (cont’d)

- The arbitrage could involve more than two currencies:
  
  ![Currency Graph]

  - 1 US Dollar
  - 2 British pound
  - 3 French franc
  - 4 German D-Mark
  - 5 Japanese Yen

If such opportunities exist, it is necessary to be able to identify them and act quickly.

- **Problem Statement**: Can a decision model be formulated to detect arbitrage opportunities in the spot currency market?
FX Arbitrage Model Overview

- **What needs to be decided?**
  - A set of spot currency transactions.

- **What is the objective?**
  - Maximize the final net amount of US dollars. (Other objectives are possible.)

- **What are the constraints? How many constraints?**
  - The final net amount of each currency must be nonnegative. For example, the total amount of all currencies converted into British pounds should be greater than the total British pounds converted into other currencies. There should be one constraint for each currency.

- **FX arbitrage model in general terms:**
  \[
  \text{max} \quad \text{Final net amount of US dollars} \\
  \text{subject to:}
  \]
  - Total currency in $\geq$ Total currency out
  - Nonnegative transactions only
**FX Arbitrage Linear Programming Model**

- **Indices:**
  Let \( i = 1, \ldots, 5 \) represent the currencies US dollar, British pound, French franc, German D-mark, and Japanese yen, respectively.

- **Decision Variables:**
  Let \( x_{ij} \) = amount of currency \( i \) to be converted into currency \( j \) (measured in units of currency \( i \)) for \( i = 1,\ldots,5 \), \( j = 1,\ldots,5 \), and \( i \neq j \).

For example, \( x_{12} \) is the number of US dollars converted into British pounds.
Cell G12: “Total out” of $ represents the total amount of $ converted to other currencies (measured in $).

Cell B17: “Total in” to $ represents the total amount of other currencies converted into $ (measured in $).

Cell B18: “Final Net in” to $ represents the final or net amount of $.
### FX Arbitrage Spreadsheet Model (continued)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FX.XLS</td>
<td>Foreign Exchange Arbitrage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Cross Currency Rates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$</td>
<td>pound</td>
<td>franc</td>
<td>mark</td>
<td>yen</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.6390</td>
<td>5.3712</td>
<td>1.5712</td>
<td>98.8901</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.5648</td>
<td>1</td>
<td>8.4304</td>
<td>2.4590</td>
<td>154.773</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.1856</td>
<td>0.1186</td>
<td>1</td>
<td>0.2921</td>
<td>18.4122</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.6361</td>
<td>0.4063</td>
<td>3.4233</td>
<td>1</td>
<td>62.9400</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.0101</td>
<td>0.00645</td>
<td>0.05431</td>
<td>0.01588</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

#### Conversion amounts

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>$</td>
<td>pound</td>
<td>franc</td>
<td>mark</td>
<td>yen</td>
<td>Total out</td>
</tr>
<tr>
<td>12</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>13</td>
<td>0.00</td>
<td>0.00</td>
<td>0.64</td>
<td>0.00</td>
<td>0.00</td>
<td>0.64</td>
</tr>
<tr>
<td>14</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>15</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>16</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>17</td>
<td>Total in</td>
<td>0.00</td>
<td>0.64</td>
<td>5.39</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>18</td>
<td>Final Net in</td>
<td>-1.00</td>
<td>0.00</td>
<td>5.39</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>19</td>
<td>Not &gt;=0</td>
<td>Not &gt;=0</td>
<td>&gt;=0</td>
<td>&gt;=0</td>
<td>&gt;=0</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Final Net $ in</td>
<td>-1.00</td>
<td>&lt;=</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Objective Function**

Cell G12: =SUM(B12:F12)

Cell B17: =SUMPRODUCT(B4:B8,B12:B16)

Cell B18: +B17-G12, Cell C18: +C17-G13, etc.

Cell B19: =IF(B18>=-0.00001,”>= 0”,“Not >= 0”)

Cell B21: +B18
The optimized spreadsheet indicates an arbitrage opportunity.

Note: Without the constraint “Final net $ in ≤ 1” the linear program would be unbounded. This constraint is needed in order for the optimizer to return a solution indicating how arbitrage profits can actually be obtained.
The optimal solution uses four currencies, US$, British pound, French franc and German D-mark. The indicated trades produce nonnegative amounts of all currencies and a positive amount of US$. Multiplying the transaction amounts by a factor $X$ would produce $X$ US$. 
FX Arbitrage Model in Algebraic Form

- Additional Decision Variables:
  Let \( f_k \) = final net amount of currency \( k \) (measured in units of currency \( k \)) for \( k = 1, \ldots, 5 \). That is, \( f_k \) is the total converted into currency \( k \) minus the total converted out of currency \( k \).

- FX Arbitrage Linear Programming Model:
  \[
  \text{max} \quad f_1 \\
  \text{subject to:}
  \]
  - Final net amount (\( f_k \)) definitions:
    \[
    f_1 = 1.5648 \ x_{21} + 0.1856 \ x_{31} + 0.6361 \ x_{41} + 0.01011 \ x_{51} - (x_{12} + x_{13} + x_{14} + x_{15}) \\
    f_2 = 0.6390 \ x_{12} + 0.1186 \ x_{32} + 0.4063 \ x_{42} + 0.00645 \ x_{52} - (x_{21} + x_{23} + x_{24} + x_{25}) \\
    f_3 = 5.3712 \ x_{13} + 8.4304 \ x_{23} + 3.4233 \ x_{43} + 0.05431 \ x_{53} - (x_{31} + x_{32} + x_{34} + x_{35}) \\
    f_4 = 1.5712 \ x_{14} + 2.4590 \ x_{24} + 0.2921 \ x_{34} + 0.01588 \ x_{54} - (x_{41} + x_{42} + x_{43} + x_{45}) \\
    f_5 = 98.8901 \ x_{15} + 154.7733 \ x_{25} + 18.4122 \ x_{35} + 62.94 \ x_{45} - (x_{51} + x_{52} + x_{53} + x_{54})
    \]
  - Bound on total arbitrage:
    \[ f_1 \leq 1 \]
  - Nonnegativity: All variables \( \geq 0 \)
Network LP?

- Is the FX arbitrage LP a *network* LP?
  
  No. Well, not quite. The flows on the arcs are multiplied by conversion rates, so it is called a network LP with *gains*. Notice also that the optimal solution is *not integer*, another indication that it is not a network linear program.

Additional Considerations

- Model needs current spot-rate data
- Live data feed and automatic solution of the linear program is highly desirable
- Typically, large transaction amounts are necessary to make significant arbitrage profits
- Similar ideas can be used to search for arbitrage opportunities in other markets
Integer Programming

Definitions. An integer program is a linear program where some or all decision variables are constrained to take on integer values only. A variable is called integer if it can take on any value in the range ..., -3, -2, -1, 0, 1, 2, 3, ... A variable is called binary if it can take on values 0 and 1 only.

What use?
- Can’t build 1.37 aircraft carriers
- Rounding may not give the best, or even a feasible, answer

Selected Applications
- Capital budgeting
  - invest all or nothing in a project
- Fixed cost/Set-up cost models
- Facility location
  - build a plant or not (yes/no decision)
- Minimum batch size
  - if any cars are produced at a plant, then at least 2,000 must be produced
  - \( C = 0 \) or \( C \geq 2,000 \) (either/or decision)
Difficulties in Solving Integer Programs

Example.

\[
\begin{align*}
\text{max} & \quad 21X + 11Y \\
\text{subject to:} & \quad 7X + 4Y \leq 13 \\
& \quad X, Y \geq 0
\end{align*}
\]

Optimal linear-programming solution: \( X = 1.83, Y = 0 \).
Rounded to \( X = 2, Y = 0 \) is infeasible.
Rounded to \( X = 1, Y = 0 \) is not optimal.
Optimal integer-programming solution: \( X = 0, Y = 3 \).
Plant-Location Problem

A new company has won contracts to supply a product to customers in Central America, United States, Europe, and South America. The company has determined three potential locations for plants. Relevant cost data are:

<table>
<thead>
<tr>
<th>Plant Locations</th>
<th>Fixed Costs</th>
<th>Variable Costs</th>
<th>Production Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>50,000</td>
<td>1,000</td>
<td>30</td>
</tr>
<tr>
<td>Philippines</td>
<td>40,000</td>
<td>1,200</td>
<td>25</td>
</tr>
<tr>
<td>Mexico</td>
<td>60,000</td>
<td>1,600</td>
<td>35</td>
</tr>
</tbody>
</table>

Fixed costs are in $ per month. Fixed costs are only incurred if the company decides to build and operate the plant. Variable costs are in $ per unit. Production capacities are in units per month. Customer demand (in units per month) is:

<table>
<thead>
<tr>
<th>Demand</th>
<th>Central America</th>
<th>United States</th>
<th>Europe</th>
<th>South America</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand</td>
<td>18</td>
<td>15</td>
<td>20</td>
<td>12</td>
</tr>
</tbody>
</table>

In addition to fixed and variable costs, there are shipping costs.
Plant-Location Problem (continued)

Numbers on arcs represent shipping costs (in $100 per unit).

Which plants and shipping plan minimize monthly production and distribution costs?
Plant-Location Model

- **Indices:**
  Let B represent the Brazil plant, and similarly use P (Philippines), M (Mexico), C (Central America), U (United States), E (Europe), and S (South America).

- **Decision Variables:** Let
  \[p_B = \text{# of units to produce in Brazil}\]
  and similarly define \(p_P\) and \(p_M\). Also let
  \[x_{BC} = \text{# of units to ship from Brazil to Central America},\]
  and define \(x_{BU}, x_{BE}, \ldots, x_{MS}\) similarly.

- **Objective Function:**
  The total cost is the sum of fixed, variable, and shipping costs. Total variable cost is:
  \[VAR = 1,000 \ p_B + 1,200 \ p_P + 1,600 \ p_M.\]
  Total shipping cost is:
  \[SHIP = 900 \ x_{BC} + 900 \ x_{BU} + 700 \ x_{BE} + 500 \ x_{BS} + 700 \ x_{PC} + 700 \ x_{PU} + 400 \ x_{PE} + 600 \ x_{PS} + 300 \ x_{MC} + 400 \ x_{MU} + 700 \ x_{ME} + 900 \ x_{MS}.\]
  We will return to the total fixed cost computation shortly.
Plant-Location Model (continued)

- **Constraints:**
  - *Plant-production definitions:* There are constraints to define total production at each plant. For example, the total production at the Mexico plant is:
    
    \[ p_M = x_{MC} + x_{MU} + x_{ME} + x_{MS} \]
    
    This can be thought of as a “flow in = flow out” constraint for the Mexico node.

- **Demand constraints:**
  - There are constraints to ensure demand is met for each customer. For example, the constraint for Europe is:
    
    \[ x_{BE} + x_{PE} + x_{ME} = 20. \]
    
    This is a “flow in = flow out” constraint for the Europe node.

- **Plant-Capacity Constraints:**
  - Production cannot exceed plant capacity, e.g., for Brazil
    
    \[ p_B \leq 30 \]


**Fixed-Cost Computation**

- *Additional Decision Variables*: To compute total fixed cost, define the *binary* plant-open variables:

  \[ y_B = \begin{cases} 
  1 & \text{if the Brazil plant is opened (i.e., if } p_B > 0) \\ 
  0 & \text{if the Brazil plant is not opened (i.e., if } p_B = 0) 
  \end{cases} \]

  and define \( y_P \) and \( y_M \) similarly.

  Total fixed cost is:

  \[ \text{FIX} = 50,000 \ y_B + 40,000 \ y_P + 60,000 \ y_M \]

  As it currently stands, the optimizer will always set the “plant open” variables to zero (so that no fixed cost will be incurred). We need constraints to enforce the meaning of these variables, e.g.,

  \[ p_B > 0 \quad \Rightarrow \quad y_B = 1. \]

  Why not add constraints to define the plant open variables, e.g., for Brazil,

  \[ y_B = \text{IF} (p_B > 0, 1, 0)? \]

  Because =IF statements are *not linear* and *discontinuous*. Optimizers cannot solve such problems easily, if at all. What else can be done?
Fixed-Cost Computation (continued)

- If $y_B = 0$ we want to rule out production at the Brazil plant. If the Brazil plant is not opened (i.e., if $y_B = 0$), its “available” capacity is 0. If $y_B = 1$, the plant is open and its “available” capacity is 30 units per month. The plant capacity constraints can be modified to enforce this meaning of $y_B$:

$$p_B \leq 30 \ y_B$$

If $y_B = 0$ then the constraint becomes $p_B \leq 0$.

If $y_B = 1$ then the constraint becomes $p_B \leq 30$.

Alternatively, if $p_B > 0$ (and $y_B$ can only take on the values 0 or 1) then $y_B = 1$.

This is exactly what is needed!

- **Modified Plant-Capacity Constraints:**

  Production cannot exceed plant capacity, e.g., for Brazil

  $$p_B \leq 30 \ y_B$$

  *Binary variable:* $y_B = 0$ or 1.

  Similar plant-capacity and binary-variable constraints are needed for the Philippines and Mexico.
Plant Location Integer Programming Model

\[
\begin{align*}
\text{min} & \quad \text{VAR} + \text{SHIP} + \text{FIX} \\
\text{Cost definitions:} & \\
(\text{VAR Def.) } & \text{VAR} = 1,000 \ p_B + 1,200 \ p_P + 1,600 \ p_M . \\
(\text{SHIP Def.) } & \text{SHIP} = 900 \ x_{BC} + 900 \ x_{BU} + 700 \ x_{BE} + 500 \ x_{BS} \\
& \quad + 700 \ x_{PC} + 700 \ x_{PU} + 400 \ x_{PE} + 600 \ x_{PS} \\
& \quad + 300 \ x_{MC} + 400 \ x_{MU} + 700 \ x_{ME} + 900 \ x_{MS} \\
(\text{FIX Def.) } & \text{FIX} = 50,000 \ y_B + 40,000 \ y_P + 60,000 \ y_M \\
\text{Plant production definitions:} & \\
(\text{Brazil) } & p_B = x_{BC} + x_{BU} + x_{BE} + x_{BS} \\
(\text{Philippines) } & p_P = x_{PC} + x_{PU} + x_{PE} + x_{PS} \\
(\text{Mexico) } & p_M = x_{MC} + x_{MU} + x_{ME} + x_{MS} \\
\text{Demand constraints:} & \\
(\text{Central America) } & x_{BC} + x_{PC} + x_{MC} = 18 \\
(\text{United States) } & x_{BU} + x_{PU} + x_{MU} = 15 \\
(\text{Europe) } & x_{BE} + x_{PE} + x_{ME} = 20 \\
(\text{South America) } & x_{BS} + x_{PS} + x_{MS} = 12 \\
\text{Modified plant capacity constraints:} & \\
(\text{Brazil) } & p_B \leq 30 \ y_B \\
(\text{Philippines) } & p_P \leq 25 \ y_P \\
(\text{Mexico) } & p_M \leq 35 \ y_M \\
\text{Binary variables: } & y_B , y_P , y_M = 0 \text{ or } 1 \\
\text{Nonnegativity: } & \text{All variables } \geq 0
\end{align*}
\]
Decision variables in cells E5:E7 are restricted to 0 or 1, i.e., they are constrained to be *binary*.

Note that many numbers in the spreadsheet were scaled to units of $100. For the optimizer to work properly, it is important (especially with integer programs) to scale the numbers to be about the same magnitude.

Shadow price information is not available with integer programs; the Excel optimizer does not give meaningful sensitivity reports.
In this spreadsheet, the plant-open cells, E5:E7, are computed with =IF statements.

The optimizer returns an incorrect optimal solution because of the =IF statements.

This is not an Excel bug. It is simply a difficult problem for any optimizer to solve because =IF statements represent discontinuous functions.
For next class

- Read pp.310-313 and Chapter 6.11 in the W&A text.