CHAPTER 5

DYNAMICS OF INVESTOR INTERACTION: AN ANALYSIS OF BSE USING AN AGENT BASED ASM MODEL

5.1 Introduction

In Chapter 2, a broad range of market structure was seen, based on the role of market participants, trading sessions, and execution mechanisms. The hardly observable processes behind price formation and behaviour of traders were also analyzed. In Chapter 3, a number of artificial stock markets (ASM) were presented. These ASM were compared based on the relevant organizational and behavioral aspects proposed in Chapter 2. LeBaron’s agent based ASM model has been chosen for carrying out the study on the dynamics of investor interaction in the BSE [63]. The empirical features generated using historical data from the BSE, will be the focus of study. Returns on stocks have traditionally been modelled by fitting a suitable statistical process to empirical returns. Studies on agent based models of stock market have been carried out by researchers, primarily on US markets [8,63,64,65,66,67,90]. Agent-based approach to stock market considers stock prices as arising from the interaction of a number of individual investors. These investors are modeled as intelligent agents, using differing lengths of past information, each trading with its own rules adapting and evolving over time, and this in turn determines the market prices.

Returns achieved on stock markets contain certain characteristic features [113,114]. These features include a distribution of returns that is more peaked than the Gaussian distribution, periods of persistent high volatility, and correlation between volatility and trading volume. It has been shown by LeBaron [64, 65, 66], that agent based models are able to demonstrate this, unlike the traditional financial models. The traditional economic models generally use either a simple distribution of returns such as the Gaussian and treat extreme events as outliers, or construct a statistical process which
reproduces some of these features. The agent-based approach considers a population of intelligent adaptive agents and let them interact in order to maximize their financial performance. It has been shown that such an approach can replicate features of real stock markets\[2,8,31,36,63,67,68,69,76,90\].

5.2 Agent Based Modeling of BSE

The methodology adopted is an extension of the work of LeBaron and aims to study the BSE by forward testing [63] an agent based model where the market is run using real data as the price input up to the current date, and then allowed to continue on into its future to enable the study of the empirical features. An attempt is made to study the behavior of agents with varying memory to see the effect on the volatility of the market. The financial data of BSE from the years 2003 to 2009 is considered for building the model, as against LeBaron’s model, where the data is entirely generated.

5.3 Review of Agent Based Model of LeBaron

The outline of the design proposed by LeBaron [63,65,66] was given in Chapter 2. In the following paragraphs, the model is further amplified for the purpose of implementation and analysis of the BSE.

5.3.1 Agent Based Model

The agents used have some “intelligence” and the investment decisions specified on a proportion of wealth are based upon an information set. A key concept of the model is that of “bounded rationality” – that an agent cannot feasibly analyse all the information in the market, and can at best combine limited “bounded” rational decision-making with empirical evidence. A Walrasian auction is adopted, wherein each agent calculates its demand for shares at every possible price and submits this to an auctioneer. The price is then set so that total demand across all agents equals total shares that can be issued.
5.3.2 Assets

The model contains two assets for investment – cash and equity. Cash pays a constant guaranteed rate of return \( r_f \) (risk free). The equity pays a dividend at each time step. This is random and the log-dividend follows a random walk:

\[
\log (d_{t+1}) = \log (d_t) + \epsilon_t
\]

where \( d_t \) is the dividend and \( \epsilon_t \) is a Gaussian random variable \( N(\mu, \sigma) \). The equity is available in a fixed supply of one share for the population. If \( s_i \) is the share holding of agent \( i \), the constraint that

\[
\sum_{i=1}^{I} s_i = 1
\]

will be always maintained, (5.2)

where \( I \) is the number of agents. The equity price arises through the interactions of the agents.

5.3.3 Agents

The model contains a number of agents. Each agent has a certain wealth and at each time step it decides how much of its wealth to consume and how much to save, and how much of its savings wealth to allocate to equity and how much to cash. The agents are of Constant Relative Risk Aversion (CRRA) of logarithmic form and at time \( t \) makes these decisions in an attempt to maximise its lifetime utility

\[
u_{i,t} = E_t \sum_{s=0}^{\infty} \beta^s \log c_{i,t+s} \]

subject to the constraint

\[
w_{i,t} = p_t s_{i,t} + b_{i,t} + c_{i,t} = (p_t + d_t) s_{i,t-1} + (1 + r_f) b_{i,t-1}
\]

where \( u_{i,t} \) is the lifetime utility of agent \( i \) from time \( t \) onwards, \( \beta \) is a constant, \( c_{i,t} \) is the consumption of agent \( i \) at time \( t \), and
$w_{i,t}$ is wealth at time $t$,

$s_{i,t}$ and $b_{i,t}$ are the risky and risk-free asset holdings,

$p_t$ the share price,

$d_t$ the dividend paid.

The two decisions, choosing $c_{i,t}$ and $s_{i,t}$, will affect the agent’s pattern of consumption. Utility would model the benefit obtained from an amount of money and utility of wealth is optimized rather than actual wealth. The optimal amount of wealth to consume at a single time step can be shown to be a constant proportion of wealth

$$c_{i,t} = (1-\beta) w_{i,t}$$

(5.5)

The time rate of discount $\beta$ is set to $1/1+r$ where $r = 0.01$ is the discount rate. The agent does not consider what happens over the whole of the future, but restricts itself only over the next single time step. In order to maximise $u_{i,t}$, it is sufficient to maximise the expected log-return:

$$E_t \log [1 + \alpha_t \cdot r_{t+1} + (1-\alpha_t) \cdot r_f]$$

(5.6)

where $\alpha_t$ is the proportion of wealth allocated to equity, $r_{t+1}$ is the return achieved from equity in the period $(t, t+1)$ and $r_f$ the constant cash return. It is not possible to perform the maximisation deductively since the equity returns distribution is not known in advance (these arise from the interaction of the agents). Therefore the agents maximise a sample expectation taken from historic returns. Because the distribution of returns may change over time, agents do not look at the whole past history, but rather look at the last $T_i$ periods, to maximise

$$\frac{1}{T_i} \sum_{K=1}^{T_i} \log [1 + (\alpha_{t-k} \cdot r_{t+1+k}) + (1-\alpha_{t-k}) r_f]$$

(5.7)

where $T_i$ is a constant for agent $i$. The choice of $T_i$ will affect an agent’s performance. There are many ways in which an agent could determine its allocation to equities. Under LeBaron’s model it does this by using one of a pool of rules.
5.3.4 Rules

A rule recommends the proportion of savings an agent should allocate to equities, taking information about the current state of the market and produces an output $\alpha \in (0,1)$. The rules are implemented as simple feed forward neural network (Figure 5.1) with a single hidden unit with restricted inputs giving an output.

![Figure 5.1: Structure of FFNN](image)

The equations given below define the network, where $z_t$ is time $t$ information and $w_j$ are parameters. $k$ takes values from 1 to 6 so that the weight array $\{w\}$ consists of 19 parameters. The output from the intermediate neuron $k$ is denoted $h_k$.

$$
\begin{align*}
    h_k &= g_1(w_{0,k} z_{t,k} + w_{1,k}) \\
    \alpha (z_t) &= g_2 \left( w_2 + \sum_{k=1}^{6} w_{3,k} h_k \right) \\
    g_1(x) &= \tanh(x) \\
    g_2(x) &= \frac{1}{2}(1+\tanh(x/2))
\end{align*}
$$

(5.8)
5.3.5 Input Values to the FFNN - The Information Set

The information set consists of six items reflecting various fundamental and technical trading strategies, and its combinations. These six items extract potentially useful information from the large quantity of historic data and simplifies the decision making process. The first three inputs are the returns on equity in the previous three time-steps, useful for technical trading. The fourth is a measure of how the current price differs from the rational-expectations price. The last two inputs measure the ratio between the current prices and exponentially weighted moving averages of the price. The Information set is:

\[ z_{t,1} = r_t = \log \left( \frac{p_t + d_t}{p_{t-1}} \right) \]

\[ z_{t,2} = r_{t-1} \]

\[ z_{t,3} = r_{t-2} \]

\[ z_{t,4} = \log(r \frac{p_t}{d_t}) \] \hspace{1cm} (5.9)

\[ z_{t,5} = \log(p_t / m_{1,t}) \]

\[ z_{t,6} = \log(p_t / m_{2,t}) \]

Where \( p_t \) is the share price, \( d_t \) is the dividend paid, \( r \) is a constant and \( m_{i,t} \) is the moving average given by

\[ m_{i,t} = \rho_i \ m_{i,t-1} + (1- \rho_i) \ p_t \] \hspace{1cm} (5.10)

with \( \rho_1 = 0.8 \) and \( \rho_2 = 0.99 \).

5.3.6 Trading and price-setting

For a given share price \( p \), each agent can determine how much of its wealth it wishes to invest and how much of this is to be invested in shares. Consequently it arrives at a demand function for shares

\[ d_{i,t}(p_t) = [ \alpha_i(p_t, I_t) \beta W_{i,t}] / p_t \] \hspace{1cm} (5.11)
where $i$ denotes the agent, $t$ refers to time, and $I_t$ represents the information set. A Walrasian auction is then used to find the price $p_t$. Walrasian auction is one in which the price is set by an auctioneer in order that the total demand for shares at that price is equal to the available supply:

$$\sum_{i=1}^{N_{\text{agents}}} d_{i,t}(p_t) = N_{\text{shares}}$$  \hspace{1cm} (5.12)

where $N_{\text{agents}}$ is the number of agents and $N_{\text{shares}}$ the number of shares. Because the new price affects the information set, and this affects the rule output which in turn affects the agents’ demand, the equations to solve are non-linear. This equation is solved using complex recursive function which searches for a value of $p_t$ that satisfies these equations starting from the price at the previous time-step.

**5.3.7 Adaptation and Evolution**

The model contains three forms of adaptation and evolution

(a) Agents can adapt by selecting a different rule. At each time step a proportion of the agents can adapt. An agent adapts by comparing the performance of its current rule with a randomly chosen rule, and switching to the new rule if the new rule scores more than the old.

(b) Agents evolve at each time step, wherein agents with the least wealth are removed from the population. It is replaced by a new agent which is given the median cash and equity holding. The new agent is given a memory length taken from a random distribution.

(c) The rules are also evolved. A rule is replaced if it has not been used for 10 time steps, and is replaced using one of three genetic operators: copying a parent and changing a single weight to a random number in (-1,1); copying a parent and adding a random number in (-0.25, 0.25) to a single weight, and copying a parent and replacing the weights for one neuron with those for the corresponding neuron of another parent.
5.4 Implementation – Agent Based Model of BSE

5.4.1 Validating the FFNN Structure for Rule

BSE SENSEX, the most popular Indian stock index has been chosen for the study. The time step considered here is one month. Prior to adopting the FFNN (Appendix B) as the structure for the Rules, it has been validated using the BSE Index data. The MATLAB Neural Network Toolbox has been chosen for creating, training and testing the network. The FFNN was trained with inputs from historical prices of BSE index, calculated taking monthly closing prices of BSE stock index from the year 2003 to 2008. The output is a simple function \(\alpha(z_t,w_j)\). The output from the network, \(\alpha\) is a 0 or 1 which would suggest where to invest in the next time-step, a 0 indicating that risk-free asset (Indian Government Public Provident Fund data considered) would give higher returns for the particular time-step and a 1 indicating that investing in an Index fund tracking the BSE would render higher returns. The network was tested with data pertaining to the year 2009 and the results have been found to validate the market scenario. It has been found that the FFNN with one hidden layer with six neurons has produced quite accurate results. Thus the FFNN has been found to establish the functional dependency between the input parameters and the market behavior and hence has been validated for the purpose of generating the rules for the agents of BSE. The neural network structure has been deliberately kept as the plain FFNN so as to retain higher degree of generality, which may be more appropriate to model invest actions.

5.4.2 Agents Modeled

The model contains of a number of agents. Each agent has properties that define its behavior. The properties of agents defined in this model are given in Table 5.1.

5.4.3 Parameter Settings

Table 5.2 gives the parameter settings used in the experiment. A total of 140 agents have been used. The model was initialised by setting agent memories and neural network parameters using a uniform random distribution over the allowable range. Each agent started with an equal shareholding set to 100.
# Table 5.1: Agent Class

<table>
<thead>
<tr>
<th>NAME</th>
<th>PROPERTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule now</td>
<td>Integer specifying the rule used by agent</td>
</tr>
<tr>
<td>Wealth</td>
<td>Wealth till (t-1) to be spent at time t</td>
</tr>
<tr>
<td>Memory</td>
<td>No of time steps agent looks back</td>
</tr>
<tr>
<td>Returns</td>
<td>Returns in the past</td>
</tr>
<tr>
<td>Proportions</td>
<td>Alpha values used in the past</td>
</tr>
<tr>
<td>Memwealth</td>
<td>Memory bound wealth</td>
</tr>
<tr>
<td>Volume</td>
<td>Volume it demands at each time step</td>
</tr>
<tr>
<td>Exist</td>
<td>No of time steps the agent has existed</td>
</tr>
</tbody>
</table>

# Table 5.2: Initializing Parameters

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Agents</td>
<td>140</td>
</tr>
<tr>
<td>Memory (Minimum to Maximum)</td>
<td>1 to 70</td>
</tr>
<tr>
<td>Length of one Time-Step</td>
<td>1 month</td>
</tr>
<tr>
<td>Number of Rules</td>
<td>250</td>
</tr>
<tr>
<td>Number of Shares</td>
<td>1</td>
</tr>
<tr>
<td>Number of time-steps projected</td>
<td>1000</td>
</tr>
<tr>
<td>Minimum Neural Network weight</td>
<td>-1</td>
</tr>
<tr>
<td>Maximum Neural Network weight</td>
<td>1</td>
</tr>
<tr>
<td>Time steps before a new rule is discarded</td>
<td>10</td>
</tr>
</tbody>
</table>
5.5 Step by Step Process

The flow diagram of the implementation is illustrated as a Step by Step Process in Figure 5.2. The simulation of the system is done for two cases:-

(a) All memory case- where agents with all memory lengths are present in the market.

(b) Long memory case- Only agents with long memory are present in the market.

The simulation runs for 1000 iterations for both the cases and the prices, returns and the interaction and evolution of agents are monitored.

5.6 Results

Forward Testing simulation was carried out by feeding data from the BSE index prices. For the first 70 months, data from BSE is used by the agents. Subsequently, the
prices are generated by Walrasian auction. The process continues for 1000 iterations. The “all-memory case” followed by “long memory case” has been implemented. The following results emerge:

(a) In the “long-memory case” where agents’ memory lengths are in the range [51,70], the prices converge after an initial learning period to the rational-expectations price, so that the returns are Gaussian. There is very little trading.

(b) In the “all-memory case” where agents’ memory lengths are in the range [1,70] prices do not settle to the rational-expectations price. Returns are more volatile, and the distribution has fat tails. Shares continue to be traded frequently.

5.7 Prices

(a) Figure 5.3 shows the variation in prices for the all memory case. The first 70 time steps show variation of BSE index prices. After the 70th time step prices are generated through Walrasian auction as a result of interaction between agents with different memory lengths and trading strategies.

(b) It can be understood that the prices vary considerably in the presence of agents with all memory lengths. The volatility in prices is evident in the above case.

(c) Figure 5.4 shows the variation in prices for the long memory case. The first 70 time steps show variation of BSE index prices. After the 70th time step prices are generated through Walrasian auction as a result of interaction between agents with similar memory lengths (long) and slightly varying trading strategies.

(d) Further, it is seen that prices tend to stabilize after some time. This is because agents converge to similar strategies over a period of time due to similar memory lengths. This in turn suppresses trading and prices stabilize to the rational expectations price. Hardly any shares are traded once the rational-expectations price has been reached. Comparing the two figures and the data from BSE, we could deduce that it is because of the presence of agents with
varying memory and different trading strategies, trading takes place. Volatility in prices arises because of these differences in agents’ properties.

5.8 Returns

(a) Logarithmic returns are considered here which is calculated from the prices and dividends. It is depending on these returns that an agent chooses his trading strategy (rules).

(b) Figure 5.5 shows variation in returns for the all memory case. The returns for the first 70 times steps are calculated using prices and dividends from the BSE. After this the returns are calculated with prices generated by the Walrasian auction and random dividends.

![Figure 5.3: Price time series for all-memory agents](image1)

![Figure 5.4: Price time series for long-memory agents](image2)
(c) Figure 5.5 clearly depicts how returns from the markets closely resemble the returns arising out of interaction between agents of varying memory lengths.

(d) Figure 5.6 shows variation in returns for the long memory case. The returns for the first 70 times steps are calculated using prices and dividends from the BSE. After this the returns are calculated with prices generated by the Walrasian auction and random dividends.

(e) The variation in returns in this case is very less compared to the actual market and the all memory case. This is because of similarity in property of agents that interact in the market. These graphs further emphasize the fact that the market is comprised of agents with different strategies, different memory length and irrational behavior.
5.9 Statistical Observations

The statistical observations obtained from the runs of the model are given at Table 5.3. The table presents summary statistics for these returns in the two different cases along with comparison of the BSE SENSEX. The first columns correspond to the series standard deviation, and the second to kurtosis. In the standard deviation, the all-memory case shows a value closer to the BSE data implying the existence of investors of all kinds of memory in a market rather than investors of only long memory. The column labeled kurtosis shows the flatness of the curve and also the peak value which in turn can give an idea about the variations from the mean.

Table 5.3: Return Summary Statistics

<table>
<thead>
<tr>
<th></th>
<th>Standard Deviation</th>
<th>Kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSE</td>
<td>4.24</td>
<td>1.9</td>
</tr>
<tr>
<td>All Memory</td>
<td>3.98</td>
<td>1.78</td>
</tr>
<tr>
<td>Long Memory</td>
<td>1.87</td>
<td>1.2</td>
</tr>
</tbody>
</table>

5.10 Other Miscellaneous Observations

Other observations detected along with possible explanations for such behavior are given below:-

(a) Agents with memory length around 25 get deleted often. Though the reason for this could not be conclusively understood, one possible explanation could be because of the fact that in the data fed in, the BSE reaches the lowest values after a period of about 25 months. Hence this could have affected the agents’ strategy, thereby affecting its performance.

(b) Agents with memory length from 50 to 60 stay for a longer period of time. Some agents even stay throughout the entire 1000 runs. A possible explanation is that this memory is the optimal period to look back in order to get the maximum returns from the market.
(c) Rules which gave a value of $\alpha$ near 0.5 were modified early. This could be attributed to two factors:–

- In cases of a fall in BSE SENSEX, rules which have a much lower $\alpha$ value reduced the losses, thus making the earlier rules poor.
- In cases of rise in Sensex, rules which have much higher $\alpha$ value increased the profits, again making the $\alpha = 0.5$ rules poor.

5.11 Variation from LeBaron Model

In order to reduce the computational complexity and to synchronize the model with the BSE, few variations have been introduced in LeBaron’s model as follows:–

(a) Memory of the agents is between [1,70], limited to a maximum of 70 months, constrained by the financial data of BSE considered from the years 2003 to 2009.

(b) Consequently, the number of agents modeled was 140 (as against 250 in LeBaron’s model). In order to have a uniform distribution of agents with different memories, two agents each with memories varying between 1 to 70 months were modeled. This choice helped in reduction of the computational complexity as well.

(c) A tolerance value of 0.1 was introduced into the Walrasian auction. To illustrate this point, in case zero was taken as the tolerance value, then comparisons of all decimal places would have to be done, leading to an infinite iteration. To surmount this, a small tolerance value has been accommodated.

(d) Since the computational complexity of ‘fzero’ function in MATLAB was very high, a customized version of the function which works in a similar manner (a simplified form of the Excel algorithm with an upper and lower bound to restrict the search) has been developed.
5.12 Future Enhancements

(a) The simulation was run for 1000 time steps due to high computational duration. It can be extended further to 10,000 runs (as suggested in the original paper) and patterns can be observed for a longer period of time. Further, owing to computational complexities, tolerance levels have been introduced, which could be removed in future simulations for better results.

(b) In the actual market, massive fluctuations occur due to natural disasters, calamities, war etc. These catastrophic events are not considered when generating prices in Walrasian auction, since in this simulation, the interaction between agents alone determines the price. Further research can be done, so as to embed theories underlying such events so as to give a more realistic approach to the price formation mechanism.

(c) Lastly, the Walrasian auction is used to generate prices in this model. Further studies could make this a prediction tool. For this to be possible, the agents may have to be redefined, their numbers and interactions increased and tested exhaustively for verification and validation.

5.13 Summary

LeBaron’s Agent Based ASM Model [63,65,66,90] has been employed to study the characteristics of BSE. This was implemented by forward testing the BSE data, a strategy different from that adopted by LeBaron, and the prices and returns were observed over a period of 1000 time steps. It has been demonstrated that the volatility as existing presently would persist, suggesting that it is indeed the interaction between agents of different strategies that brings volatility and trading in the market. In order to bring about stability (thereby bringing about “Healthy” volatility), it is imperative that the financial structure of the BSE is revisited. A possible solution could be the introduction of market-makers, as is prevalent in the more mature markets of NASDAQ or NYSE. Introduction of a market maker will play an important role in the price formation process.