FORECASTING DAILY VOLATILITY OF FOREIGN EXCHANGE MARKETS - A COMPARISON OF THE ARCH MODEL AND A NEW MODEL USING HIGH FREQUENCY DATA

by

YE-HSIANG LAI

B.B.A., Business Administration
National Chengchi University
(1988)

Submitted to the Sloan School of Management
in Partial Fulfillment of
the Requirements of the Degree of
Master of Science in Management

at the

Massachusetts Institute of Technology
February 1993

Signature of Author ______________________________
MIT Sloan School of Management
January 26, 1993

Certified by ______________________________
Bin Zhou
Assistant Professor of Statistics and Management Science
Thesis Supervisor

Accepted by ______________________________
Jeffrey A. Barks
Associate Dean, Master's and Bachelor's Programs

FORECASTING DAILY VOLATILITY OF FOREIGN EXCHANGE MARKETS - A COMPARISON OF THE ARCH MODEL AND A NEW MODEL USING HIGH FREQUENCY DATA

by

YE-HSIANG LAI

Submitted to the Alfred P. Sloan School of Management
on January 26, 1993 in partial fulfillment
of the requirements of the Degree of
Master of Science in Management

ABSTRACT

The phenomenal growth of derivative markets has made understanding security price volatility just as important as understanding their actual price levels. The reason is that the pricing of derivative products is highly dependent on the expected volatility in price of the underlying asset during the term of the option contract. Although volatility can be estimated from low frequency data using model such as ARCH (Engle, 1982), the estimates are often lagged and not very accurate. Recent availability of high frequency data offers us a new opportunity to study volatility. This thesis compares the forecasting performance of the ARCH model and the model using high frequency data. We demonstrate the superiority of the model using high frequency data to ARCH in forecasting the daily volatility.

Thesis Supervisor: Bin Zhou
Title: Assistant Professor of Statistics and Management Science

TABLE OF CONTENTS

1. Introduction

1.1 The Importance of Volatility Forecasting

Understanding the behavior of volatility plays a crucial role in financial markets. The importance is even enhanced by the huge growth of derivative products in financial markets during the 1980s. The ability to better forecast volatility is of extreme importance for determining more accurate prices of the derivative products.

In 1973, Fischer Black and Myron Scholes developed the famous option pricing formula (Black and Scholes 1973). This formula shows us the relationship between an option's price and several other factors, including the volatility of the underlying asset's price. According to this formula, all option prices are a function of the expected volatility of the underlying assets during the duration of the options life.

In order to derive an option's price from Black-Scholes formula, the option is replicated by a portfolio consisting of the underlying asset and a risk-free bond. The initial portfolio depends on the same variables necessary to price the option, including expected future volatility. This replication allows for an arbitrage opportunity if the option's market price is different from the initial cost of the portfolio. A different expectation of volatility will result in a different option price from this formula. Therefore, an option trader is able to make profit by making superior forecasts of future volatility.

With such an enormous portion of the financial markets so dependent on the volatility behavior of financial markets, there are obvious benefits to understanding the volatility better. Improved forecasts of volatility would allow traders to price their options more accurately.

Low frequency data in general only give us high, low, open, and close prices each day. It is difficult to understand the volatility dynamics and to estimate the magnitude of daily volatility by such data since the volatility changes over time. However, high frequency data gives us potential for studying volatility in details because it provides us with a nearly continuous observation of the price process.

It can be shown that better estimation of volatility is made by using more frequent data, rather than by data collected over a longer period, since the assumption of stationarity may not hold. Parameters used to model short-term volatility may not be suitable for modeling long-term volatility.

1.2 Existing Volatility Models

Various models of estimating volatility have been proposed by Clark (1973), Engle (1982), Tauchen and Pitts (1983), Taylor (1986), and Bollerslev (1986). Suppose a daily return process X_t is normally distributed condition on volatility $V_t = v_t$:

(1) $$(X_t | V_t = v_t) \sim N(\mu, v_t)$$

where $\{V_t\}$ is the market volatility.

Then volatility $\{V_t\}$ can be forecasted by the following methods:

1) forecasting future volatility by historical standard deviation assuming that the volatility is stable:

The accuracy of this method is limited in the following ways: 1) the number of samples used to estimate the past volatility is a parameter that is difficult to determine, and 2) the method assumes that there will be no change in the future volatility, an unrealistic assumption for the market.

2) modeling volatility by exogenous variables:

Linking variance changes on exogenous variables is another way to model the $\{V_t\}$ process. Many events could cause changes in the market volatility. For example, Clark (1973) suggested that V_t could be a function of trading volume. Tauchen and Pitts (1983) related V_t to the number of relevant information items during day t. However, the relation between the volatility and exogenous variables is complex and difficult to handle.

3) modeling volatility by past prices:

This is an extreme alternative to 2), namely that V_t is a function of past returns. One famous example is Engle's Autoregressive Conditional Heteroscedasticity (ARCH) model. An ARCH model can be written as

$$f(\omega_t) = \alpha_0 + \sum_{i=1}^{p} \alpha_i (X_{t-i} - \mu)^2, \quad \alpha_0 > 0, \quad \alpha_i \geq 0, \; i = 1, \cdots, p$$

(A more in-depth description of ARCH is given in the next section.) Engle's ideas are important because they offer potential models for changing conditional variances and catch the dynamic structure of the volatility process. This model can be assessed by using observed data. After ARCH, other similar models were also introduced, including Generalized ARCH (GARCH) model by Bollerslev (1986) and the ARMACH model by Taylor (1986), which combines ARCH and the Autoregressive Moving Average (ARMA) model.

1.3 The Objective of This Thesis

The computing power of modern information technology makes the accumulation of high frequency data possible. Tick-by-tick data give us a picture of a nearly continuous observation of the markets and offer us the opportunity to study volatility in a new way.

Using the high frequency data of foreign exchange markets, Zhou (1992) proposed a method of volatility estimation. In this thesis, I use those estimated volatilities and apply the ARMA model to forecast volatility. The results are compared with those by ARCH. I find that by using high frequency data, I am able to better forecast daily volatility than by using ARCH model. However, as I expected, the superiority to ARCH decreases as the number of forecasting steps increases.

2. ARCH Model

Consider a random variable y_t which is drawn from the conditional density function $f(y_t|y_{t-1},\cdots)$. The forecast of the value at time t, based on the past information, under standard assumptions, is $E(y_t|y_{t-1},\cdots)$. The variance of this one-period forecast is given by $V(y_t|y_{t-1},\cdots)$. Before ARCH was introduced, for the conventional econometric models, the conditional variance does not depend on $(y_{t-1},\cdots)$. Engle proposed ARCH model where the variance depends on the past realization of the series. A simple case is

(2) $$y_t = \varepsilon_t h_t^{1/2},$$

(3) $$h_t = \alpha_0 + \alpha_1 y_{t-1}^2,$$

with $\mathrm{var}(\varepsilon_t) = 1$. Adding the assumption of normality, it can be more directly expressed in terms of ψ_t, the information set available at time t. Using conditional densities,

(4) $$y_t|\psi_{t-1} \sim N(0,h_t),$$

$$h_t = \alpha_0 + \alpha_1 y_{t-1}^2,$$

The variance function can be expressed more generally as

(5) $$h_t = h(y_{t-1}, y_{t-2}, \ldots, y_{t-p}, \boldsymbol{a})$$

where p is the order of the ARCH process and $\boldsymbol{a}$ is a vector of unknown parameters.

To decide how many parameters in $\boldsymbol{a}$ are the best for modeling our foreign exchange market data, the AIC criterion is used. AIC is defined by Akaike (1973) as follows:

(6) AIC = -(maximum likelihood function) + 2×(number of parameters).

By this criterion, the number of parameters with the minimum AIC is the best.

Let l be the summation of log-likelihood and l_t be the log-likelihood of the tth observation and T the sample size. Then

(7) $$l = \sum_{t=1}^{T} l_t,$$

(8) $$l_t = -\frac{1}{2}\log h_t - \frac{1}{2}y_t^2 / h_t.$$

Suppose the h function is pth order linear (in the squares), so that it can be written as

(9) $$h_t = \alpha_0 + \alpha_1 y_{t-1}^2 + \cdots + \alpha_p y_{t-p}^2.$$

Let $z_t' = (1, y_{t-1}^2, \cdots, y_{t-p}^2)$ and $\boldsymbol{a}' = (\alpha_0, \alpha_1, \cdots, \alpha_p)$, so that (9) can be rewritten as

(10) $$h_t = z_t' \boldsymbol{a}.$$

The maximum likelihood estimate of $\boldsymbol{a}$ can be obtained by solving the following equation:

(11) $$\frac{\partial l}{\partial \alpha} = \sum_{t=1}^{T} \frac{1}{2h_t} z_t \left(\frac{y_t^2}{h_t} - 1 \right) = 0.$$

The ARCH model has several advantages in forecasting volatility. ARCH captures the clustering attribute of volatility. Because ARCH uses observed and well-collected data (past prices), it is easy to be handled mathematically. Some empirical studies (Engle 1980, 1982) have claimed ARCH can make close approximation to many financial time series.

A C-language program has been designed to solve (11) by using Newton-Raphson method and to compute the corresponding maximum likelihood estimate. (A printout of the program is attached in Appendix.)

3. Estimating Daily Volatility by High Frequency Data

Many large financial institutions have collected "tick-by-tick" foreign exchange quotes. The usefulness of tick-by-tick data in estimating volatility was brought to attention by Zhou (1992).

Zhou found that level of noises is a fundamental difference between the high and low frequency data. The noise is negligible in low frequency data, such as daily and weekly data, but becomes very significant in high frequency "tick-by-tick" data. The noise may come from many different sources, such as round-off error, updating noises, and typographical errors. Summarizing all these arguments, Zhou assumed the following process for the exchange rate:

(12) $$S(t) = d(t) + B(\tau(t)) + \varepsilon_t.$$

In (12), $S(t)$ is logarithm of the exchange rate. $B(\cdot)$ is the standard Brownian motion. Both $d(\cdot)$ and $\tau(\cdot)$ are assumed to be deterministic functions. $\tau(\cdot)$ has positive increments. ε_t is the mean zero random noise independent to the Brownian motion $B(\cdot)$.

Suppose that we have observations $\{S(t_i), i = -2k, -2k+1, \cdots, n\}$ from process (12). Let $X_{i,k} = S(t_i) - S(t_{i-k})$. Given time $[0, n]$, the function $\tau(t)$ can be estimated incrementally. Zhou showed that the volatility $\tau(t_n) - \tau(t_0)$ can be estimated by:

(13) $$V(t_0, t_n) = \frac{1}{k} \sum_{i=1}^{n} (X_{i,k}^2 + 2X_{i,k} X_{i-k,k}).$$

In Zhou's study, the tick-by-tick data of foreign exchange quotations for the entire year of 1990 are used. Summary statistics of these 1990 tick-by-tick returns are listed in Tables 1 and 2. The average return of the tick-by-tick data is negligible in comparison to its standard deviation.

0:00 and 24:00 Greenwich Mean Time (GMT) are chosen as the beginning and ending hours of a trading day because this 24-hour period covers most foreign exchange trading activities in the world: 0:00 GMT is 9:00 a.m. Tokyo time and 24:00 GMT is 7:00 p.m. New York time.

By formula (13), the 1990 volatility of the DM/$ return is computed and plotted in Figure 1. Figures 2 and 3 illustrate the Q-Q plots of the 1990 daily volatility and its natural logarithms (log-volatility). Figures 2 and 3 show that log-volatility fits normal distribution much better than volatility does.

4. Comparison of ARCH Model and the Model Using High Frequency Data

4.1 Introduction

After getting the estimates of daily volatility in 1990 by using high frequency data, ARMA model, a well-regarded time series model developed by Box and Jenkins (1970), is used to forecast the 1991 daily volatility on business days from January 1, 1991 to June 30, 1991. The ARCH model is used as well to forecast 1991 daily volatility of the same period. I compare the forecasted volatilities of these models with the estimated volatilities and implied volatilities. Due to far less trading on Saturday and Sunday GMT than on week days, the weekend data are excluded from my data set in my modeling and forecasting analysis.

4.2 Parameter Estimation of ARCH Model

To forecast the daily volatility of year 1991, I use 1990 daily data to estimate the parameters in the ARCH model. From Table 1, we know that the average return of the tick-by-tick data is negligible in comparison to its standard deviation. Therefore, when using ARCH to model 1990 volatility, I assume the mean of DM/$ daily return is 0. It simplifies our computation without tangibly affecting the result.

In this thesis, y_t in equation (5) is the continuously compounded return of DM/$ on day t multiplied by 100. In other words,

$$(14) \qquad y_t = \ln\left(\frac{S(t)}{S(t-i)}\right) \times 100,$$

where $S(t)$ is the DM/$ exchange rate at time t.

The results can be found in Table 3 and show that ARCH(3) is the best for modeling the 1990 data. The parameter values are

α_0=0.376486,

α_1=-0.004612,

α_2=0.061097,

α_3=-0.000355.

Because α_1 and α_3 are smaller than 0, it is possible to get negative volatility. But the absolute values of these two parameters are small compared to α_0 (constant), the chance of getting negative volatility is very small. Therefore, we decide to use these parameters to forecast volatility. In case there is any negative volatility, we can replace it with zero; however, such cases have not appeared in our forecasting.

The next step is to forecast the daily volatility by these ARCH parameters. The 1-step, 2-step, and 5-step forecasting formulae are given as follows:

1-step: $\hat{h}_t^{(1)} = \alpha_0 + \alpha_1 y_{t-1}^2 + \alpha_2 y_{t-2}^2 + \alpha_3 y_{t-3}^2$

2-step: $\hat{h}_t^{(1)} = \alpha_0 + \alpha_1 \hat{h}_{t-1}^{(1)} + \alpha_2 y_{t-2}^2 + \alpha_3 y_{t-3}^2$

5-step: $\hat{h}_t^{(1)} = \alpha_0 + \alpha_1 \hat{h}_{t-1}^{(1)} + \alpha_2 \hat{h}_{t-2}^{(1)} + \alpha_3 \hat{h}_{t-3}^{(1)}$

where y_{t-p} is the continuously compounded return of DM/\$ p days ago multiplied by 100. $\hat{h}_t^{(1)}/10000$ is the expected daily variance of DM/\$ returns.

4.3 ARMA Model

With Zhou's estimated daily volatilities of 1990 DM/\$, I use ARMA model to parameterize the 1990 volatility. A general ARMA process is defined by using p autoregressive and q moving-average parameters:

$$X_t - \mu = \sum_{i=1}^{p} a_i (X_{t-i} - \mu) + \sum_{j=0}^{q} b_j \varepsilon_{t-j} \tag{15}$$

with $b_0 = 1$, $a_p \neq 0$, and $b_p \neq 0$.

To choose the appropriate number of AR parameters p and MA parameters q, RATS 3.01 time series software is used to model both 1990 estimated volatilities and log-volatilities. (The results are listed in Table 4). With the higher $\overline{\mathrm{R}}^2$ value, ARMA model fits the log-volatility data better than it fits the volatility data. The parameter values of ARMA(2,1), ARMA(2,0), ARMA(3,0), and ARMA(4,0) are listed in Tables 5 and 6.

In order to compare with ARCH model, which does not have moving-average parameter, in addition to ARMA(2,1), the autoregressive models ARMA(2,0), ARMA(3,0), and ARMA(4,0) are all used to forecast the 1991 daily volatility. Both volatility and log-volatility data are used in forecasting. . The forecasting formulae are given as follows:

- ARMA(2,1)

1-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 h_{t-1} + \text{AR}_2 h_{t-2} + \text{MA}_1 \varepsilon_{t-1}$

2-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 \hat{h}_{t-1}^{(2)} + \text{AR}_2 h_{t-2}$

5-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 \hat{h}_{t-1}^{(2)} + \text{AR}_2 \hat{h}_{t-2}^{(2)}$

where $\hat{h}_t^{(2)}$ is the expected daily variance of DM/$ returns.

- ARMA(2,0)

1-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 h_{t-1} + \text{AR}_2 h_{t-2}$

2-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 \hat{h}_{t-1}^{(2)} + \text{AR}_2 h_{t-2}$

5-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 \hat{h}_{t-1}^{(2)} + \text{AR}_2 \hat{h}_{t-1}^{(2)}$

- ARMA(3,0)

1-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 h_{t-1} + \text{AR}_2 h_{t-2} + \text{AR}_3 h_{t-3}$

2-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 \hat{h}_{t-1}^{(2)} + \text{AR}_2 h_{t-2} + \text{AR}_3 h_{t-3}$

5-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 \hat{h}_{t-1}^{(2)} + \text{AR}_2 \hat{h}_{t-2}^{(2)} + \text{AR}_3 \hat{h}_{t-3}^{(2)}$

- ARMA(4,0)

1-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 h_{t-1} + \text{AR}_2 h_{t-2} + \text{AR}_3 h_{t-3} + \text{AR}_4 h_{t-4}$

2-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 \hat{h}_{t-1}^{(2)} + \text{AR}_2 h_{t-2} + \text{AR}_3 h_{t-3} + \text{AR}_4 h_{t-4}$

5-step: $\hat{h}_t^{(2)} = \text{constant} + \text{AR}_1 \hat{h}_{t-1}^{(2)} + \text{AR}_2 \hat{h}_{t-2}^{(2)} + \text{AR}_3 \hat{h}_{t-3}^{(2)} + \text{AR}_4 \hat{h}_{t-4}^{(2)}$

4.4 Comparison:

Zhou's estimated daily volatility (VOLDAY) as well as implied daily volatility (IMPLIED), obtained from J. P. Morgan, are used as "true volatility" in the forecasting performance comparison of ARCH and the model using high frequency data. Since the distribution of the log-volatility is more close to normal distribution than the distribution of the volatility itself, mean squared log-difference (MSLD) is used to evaluate their forecasting performance.

$$\text{MSLD} = \frac{1}{n}\sum_{t=1}^{n}(\ln \hat{h}_t - \ln h_t)^2,$$

where $\hat{h}_t$ is the forecast of future volatility and h_t is "true volatility". Denote $\varepsilon_t = \ln \hat{h}_t - \ln h_t$. I further decompose MSLD into bias and variation:

$$\text{MSLD} = (\text{E}(\varepsilon))^2 + \text{var}(\varepsilon).$$

Generally speaking, MSLD measures the total forecasting performance. (The smaller, the better.) $(\text{E}(\varepsilon))^2$ points out the forecasting bias from the "true volatility". $\text{var}(\varepsilon)$ captures the model's ability to trace the volatility movement. We are especially interested in the magnitude of MSLD and $\text{var}(\varepsilon)$.

The results of our computation are listed in Tables 7, 8, and 9. The numbers in parentheses are ratios of MSLD of the model using high frequency data against those of ARCH.

The result is interesting. The new models, both ARMA on log-volatility and ARMA on volatility, demonstrate much superiority to ARCH in forecasting. When VOLDAY is used as "true volatility", the ARMA model on volatility has better performance than ARMA on log-volatility. From Table 4, I find that ARMA on log-volatility fits the data better than ARMA on volatility does. But we know that forecasting is different from modeling. The volatility dynamics of the past may be different from those of the future.

When VOLDAY is used as "true volatility", ARMA on volatility has MSLD 51% ~ 55% in 1-step forecast, 70% ~ 75% in 2-step, and 80% ~ 92% in 5-step, of those using

ARCH. As the number of steps increases, it seems that the performance of the new models and ARCH converge to the same level. It is not surprising because when making long-range forecasts, most models only give us the estimate of the average level of volatility.

When IMPLIED is used as "true volatility", ARMA on log-volatility has better performance than ARMA on volatility. It has MSLD of 59% ~ 71% in 1-step forecast, 61% ~ 64% in 2-step, and 71% ~ 83% in 5-step, of those using ARCH. Again, the superiority of the new model decreases as the number of steps increases.

Further, from Figures 4 and 5, the superior performance of the new model in comparison to ARCH is clear. These figures show that both ARCH and the new model have lagged effect of forecasting. But the new model traces the changing trend of VOLDAY or IMPLIED much better than ARCH does, which only makes small changes around the average at all times.

5. Conclusion

High frequency data does not only provide us with the opportunity to improve estimation of daily volatility but also to better forecast the future volatility. In terms of the 1-step and the 2-step forecasting, its performance is phenomenal. For example, both ARMA(3,0) and ARCH have 4 parameters (including constant) in this study. When VOLDAY is used, the MSLD of ARMA(3,0) on volatility are only 52% and 72% of ARCH's in the 1-step and the 2-step forecasts, respectively.

The convergence of these two models' forecasting performance in the long-term was expected. In the long-term forecast, basically, all models can only forecast on average. If we want to have a better long-term forecast using daily volatility, long memory process models or new time series models are needed to capture the long-term fluctuation.

Tables

Table 1: Summary Statistics of Tick-by-tick Returns

n	=	2129364	Med.	=	.000000	Skew.	=	-0.0399
Mean	=	-5.5486e-8	Min.	=	-.006621	Kurt.	=	10.7544
S.D.	=	2.3970e-4	Max.	=	0.007515	ρ_1	=	-0.4636

Table 2: Average Daily Volatility of 1990 DM/$ Return

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Ave.Vol.	7.54e-6	3.25e-5	4.13e-5	3.63e-5	4.45e-5	4.12e-5	7.07e-8
SE	1.11e-6	2.15e-6	4.42e-6	2.43e-6	4.39e-6	3.92e-6	3.19e-8
n	50	51	50	51	51	51	37

Table 3: ARCH Maximum Likelihood Estimates and AIC Values

Number of Parameters	2	3	4	5
Max. Likelihood Estimate	-22.05395	-15.23068	**-9.822805**	-7.877764
AIC	26.05	21.23	**17.82**	17.88

Table 4: $\overline{R}^2$ Values of the ARMA Model

Modeling Volatility		
p	q	$\overline{R}^2$
1	0	0.2914
2	0	0.2355
2	1	0.2963
2	2	0.2937
3	0	0.2321
4	0	0.2310
5	0	0.2051

Modeling Log-Volatility		
p	q	$\overline{R}^2$
1	0	0.3008
2	0	0.3356
2	1	0.3704
2	2	0.3692
3	0	0.3314
4	0	0.3346
5	0	0.3232

Table 5: Parameter Values of ARMA on Log-Volatility

	(2,1)	(2,0)	(3,0)	(4,0)
Constant	-10.41942	-10.33020	-10.33577	-10.35003
AR_1	1.191642	0.4245517	0.4066887	0.3924353
AR_2	-0.2288341	0.2274615	0.2073665	0.1705100
AR_3	N/A	N/A	0.0528016	0.0037908
AR_4	N/A	N/A	N/A	0.1344153
MA_1	-0.8362858	N/A	N/A	N/A

Table 6: Parameter Values of ARMA on Volatility

	(2,1)	(2,0)	(3,0)	(4,0)
Constant	3.469228e-5	3.868503e-5	3.846604e-5	3.760480e-5
AR_1	1.206383	0.4168307	0.4010656	0.3797254
AR_2	-0.2483219	0.1252296	0.0996073	0.0687916
AR_3	N/A	N/A	0.0695464	0.0422969
AR_4	N/A	N/A	N/A	0.1024739
MA_1	-0.8829202	N/A	N/A	N/A

Table 7: 1-Step Forecast

VOLDAY			
ARCH			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
	0.352592	0.059562	0.293030
ARMA on Log-Volatility			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
(2,1)	0.200601 (0.57)	0.019163	0.181438 (0.62)
(2,0)	0.208036 (0.59)	0.028714	0.179322 (0.61)
(3,0)	0.205592 (0.58)	0.026903	0.178689 (0.61)
(4,0)	0.196672 (0.56)	0.023086	0.173586 (0.59)
ARMA on Volatility			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
(2,1)	0.193848 (0.55)	0.003928	0.189919 (0.65)
(2,0)	0.186478 (0.53)	0.008204	0.178274 (0.61)
(3,0)	0.183603 (0.52)	0.006405	0.177198 (0.60)
(4,0)	0.180388 (0.51)	0.005679	0.174708 (0.60)

IMPLIED			
ARCH			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
	0.074825	0.005786	0.069040
ARMA on Log-Volatility			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
(2,1)	0.040846 (0.55)	0.000026	0.040820 (0.59)
(2,0)	0.049401 (0.66)	0.000712	0.048689 (0.71)
(3,0)	0.046656 (0.62)	0.000506	0.046150 (0.67)
(4,0)	0.045697 (0.61)	0.000119	0.045578 (0.66)
ARMA on Volatility			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
(2,1)	0.049855 (0.67)	0.006523	0.043332 (0.63)
(2,0)	0.056013 (0.75)	0.002524	0.053489 (0.77)
(3,0)	0.054110 (0.72)	0.003399	0.050710 (0.73)
(4,0)	0.053936 (0.72)	0.003872	0.050064 (0.73)

Table 8: 2-Step Forecast

VOLDAY			
ARCH			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
	0.344054	0.060409	0.283644
ARMA on Log-Volatility			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
(2,1)	0.264788 (0.77)	0.035444	0.229344 (0.81)
(2,0)	0.286683 (0.83)	0.059370	0.158353 (0.56)
(3,0)	0.279281 (0.81)	0.054256	0.225026 (0.76)
(4,0)	0.263626 (0.77)	0.045523	0.218103 (0.79)
ARMA on Volatility			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
(2,1)	0.254374 (0.74)	0.011970	0.242404 (0.85)
(2,0)	0.256885 (0.75)	0.024664	0.232221 (0.82)
(3,0)	0.247476 (0.72)	0.019415	0.228061 (0.80)
(4,0)	0.239320 (0.70)	0.017005	0.222315 (0.78)

IMPLIED			
ARCH			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
	0.071494	0.005227	0.066267
ARMA on Log-Volatility			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
(2,1)	0.042101 (0.59)	0.001435	0.040666 (0.61)
(2,0)	0.051113 (0.71)	0.008518	0.042595 (0.64)
(3,0)	0.048267 (0.68)	0.006946	0.041321 (0.62)
(4,0)	0.044810 (0.63)	0.004302	0.040508 (0.61)
ARMA on Volatility			
	MSLD	$(E(\varepsilon))^2$	$var(\varepsilon)$
(2,1)	0.044300 (0.62)	0.001819	0.042482 (0.64)
(2,0)	0.042339 (0.59)	0.000053	0.042286 (0.64)
(3,0)	0.041003 (0.57)	0.000047	0.040956 (0.62)
(4,0)	0.042049 (0.59)	0.000191	0.041858 (0.63)

Table 9: 5-Step Forecast

VOLDAY				IMPLIED			
ARCH				**ARCH**			
	MSLD	$(E(\varepsilon))^2$	var(ε)		MSLD	$(E(\varepsilon))^2$	var(ε)
	0.396957	0.086475	0.310482		0.077615	0.012283	0.065331
ARMA on Log-Volatility				**ARMA on Log-Volatility**			
	MSLD	$(E(\varepsilon))^2$	var(ε)		MSLD	$(E(\varepsilon))^2$	var(ε)
(2,1)	0.339775 (0.86)	0.063210	0.276565 (0.89)	(2,1)	0.061754 (0.80)	0.007742	0.054012 (0.83)
(2,0)	0.435946 (1.10)	0.158353	0.277593 (0.89)	(2,0)	0.100289 (1.29)	0.050916	0.049373 (0.76)
(3,0)	0.417923 (1.05)	0.145288	0.272635 (0.88)	(3,0)	0.092532 (1.19)	0.044586	0.047945 (0.73)
(4,0)	0.375878 (0.95)	0.113269	0.262610 (0.85)	(4,0)	0.076208 (0.98)	0.029762	0.046446 (0.71)
ARMA on Volatility				**ARMA on Volatility**			
	MSLD	$(E(\varepsilon))^2$	var(ε)		MSLD	$(E(\varepsilon))^2$	var(ε)
(2,1)	0.317982 (0.80)	0.026922	0.291060 (0.94)	(2,1)	0.056687 (0.73)	0.000028	0.056659 (0.87)
(2,0)	0.364545 (0.92)	0.076879	0.287666 (0.93)	(2,0)	0.065436 (0.84)	0.010377	0.055059 (0.84)
(3,0)	0.343249 (0.86)	0.064927	0.278322 (0.90)	(3,0)	0.059076 (0.76)	0.006987	0.052089 (0.80)
(4,0)	0.322294 (0.81)	0.053321	0.268973 (0.87)	(4,0)	0.054449 (0.70)	0.004377	0.050072 (0.77)

Figures

Figure 1: 1990 Daily Volatility of the DM/$ Returns

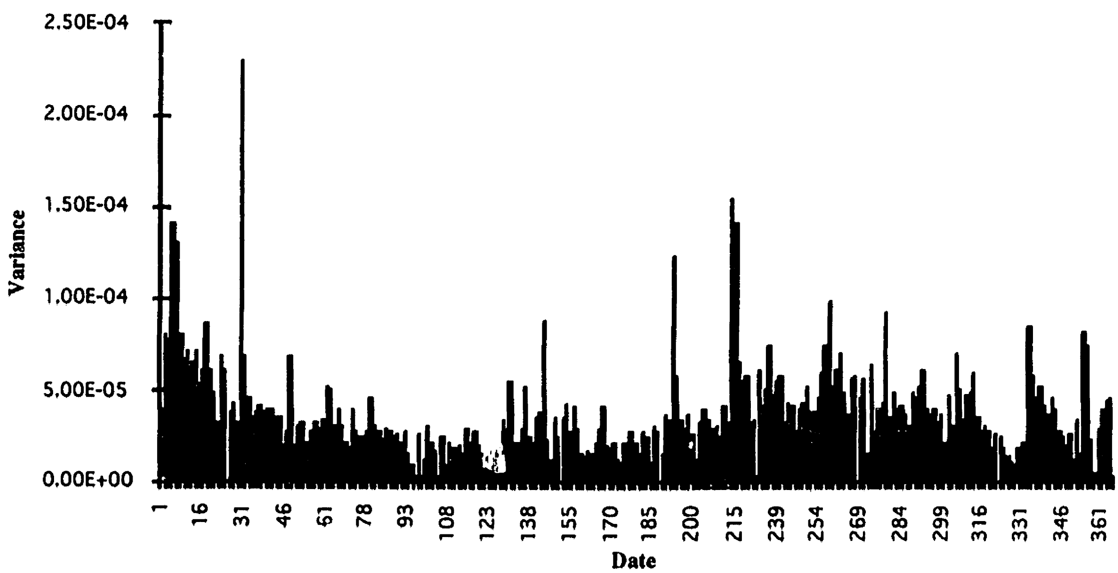


Figure 2: Q-Q Plot of 1990 Daily Volatility

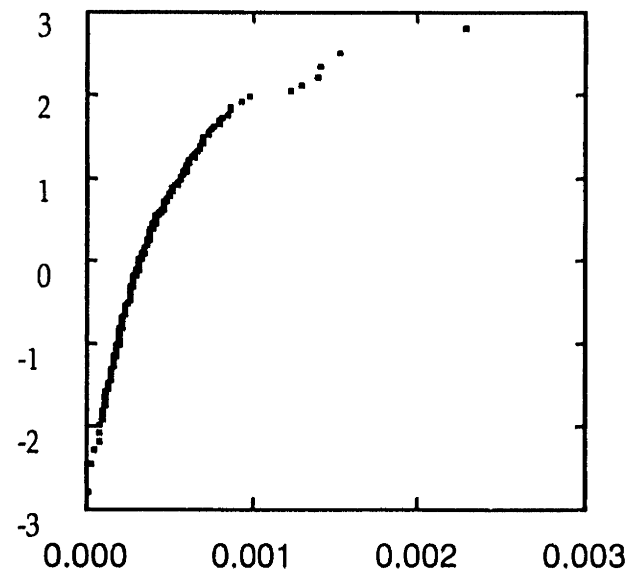


Figure 3: Q-Q Plot of 1990 Daily Log-Volatility

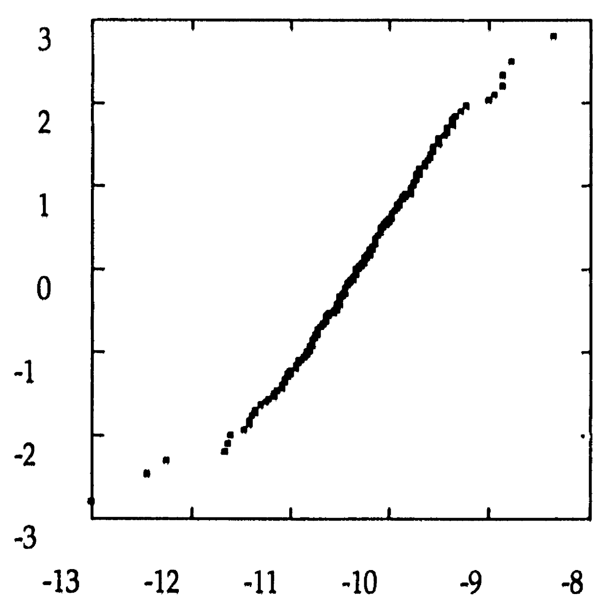


Figure 4: Comparison of ARCH and ARMA(2,1) Using High Frequency Data in 1-step Forecasting ("True Volatility": VOLDAY)

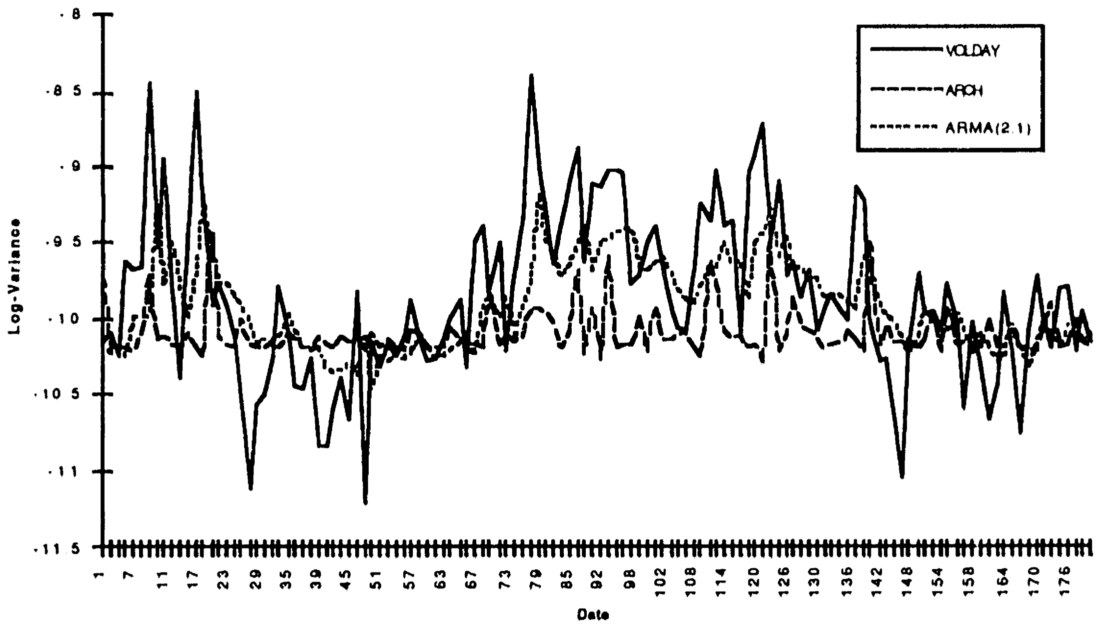
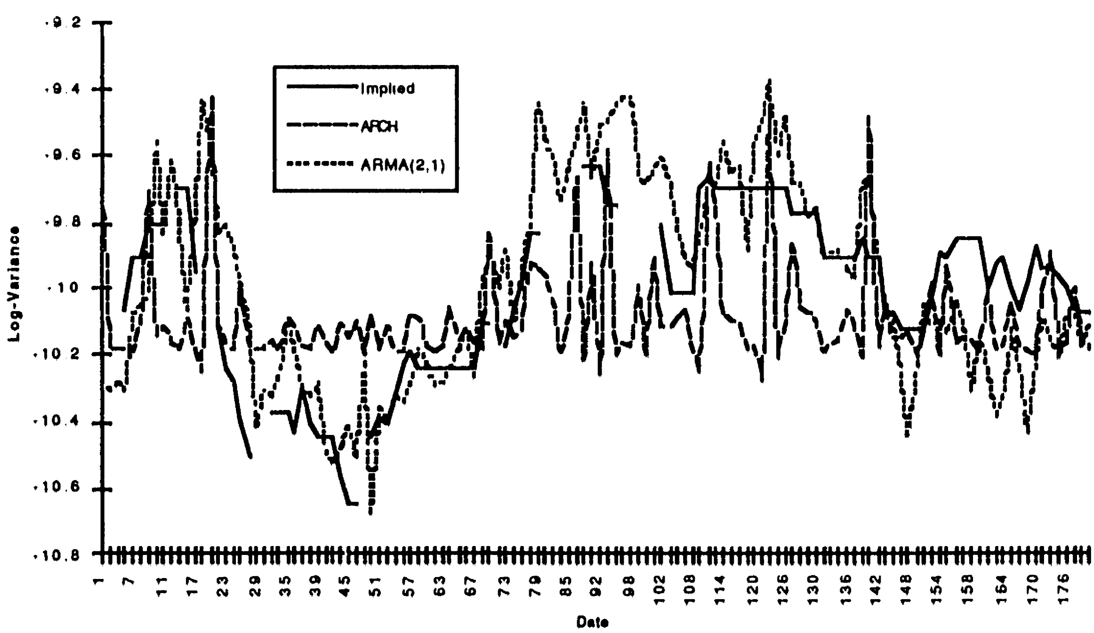


Figure 5: Comparison of ARCH and ARMA(2,1) Using High Frequency Data in 1-step Forecasting ("True Volatility": IMPLIED)



APPENDIX

```
/* ARCH.c
    This program uses Newton-Raphson method subroutine in Numerical Recipes to
    compute the ARCH likelihood function value under a designated parameter number
    (P) and derive their specific values.  It retrieves continuously compounded return
    data directly.
*/

#define TINY 1.0e-20;
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define P 6
#define SIZE 255    /* Size: number of record*/
float  yhl_yt[SIZE];

main()
{
 FILE *result_file;
 int i;
   float x[10];
   double L;
        x[1]=.4;
        for (i=2;i<=P;i++)
        {x[i]=.0;}
   result_file=fopen("result.dat","w");
        get_yt(yhl_yt,"DM.w/1");
        mnewt(100,x,P,.0000001,.0000001);
     for (i=1;i<=P;i++)
        {fprintf(stderr, "alpha%d=%f\n",i-1,x[i]);
         fprintf(result_file, "alpha%d=%f\n",i-1,x[i]);}
   likelihood(x,&L);
    printf("Likelihood=%f\n",L);
}

/* Retrieve log-normal return data */
get_yt(y,fname)
float *y;
char *fname;
{
  FILE *fp;
  int rc;
  fp = fopen(fname, "r");
  if (fp == (FILE *)NULL) {
        fprintf(stderr, "can't read the file=%s\n", fname);
        exit(0);
  }
  rc=fscanf(fp,"%*d %*d %*d %*d %*f %*f %f\n", y);
  while (rc!=EOF)
  {
```

```
  y++;
  rc=fscanf(fp,"%*d %*d %*d %*d %*f %*f %f\n". y);
 }
 return;
}


/* Give user function to Newton-Raphson */
void usrfun(x,alpha,bet)
float *x;
float **alpha, *bet;
{
        float Ht,y0,y1,z[10];
        int i,j,k;

      for (i=1;i<=P;i++)
        {
          bet[i]=0.0;
          for (j=1;j<=P;j++)
          { alpha[i][j]=0.0; }
        }

        for(i=P-1;i<SIZE;i++)
        {
            z[0]=1;
            for(j=1;j<P;j++) z[j]=yhl_yt[i-j]*yhl_yt[i-j];
            y0=yhl_yt[i]*yhl_yt[i];
            Ht=0;
            for(j=1;j<=P;j++) Ht+=x[j]*z[j-1];
                  if (Ht<.01) Ht=.01;

                for (j=1;j<=P;j++)
                   {bet[j] -= (.5/Ht)*z[j-1]*(y0/Ht-1.0);}

        for (j=1;j<=P;j++){
          for (k=1;k<=P;k++)
              alpha[j][k]+=(.5-y0/Ht)*z[j-1]*z[k-1]/Ht/Ht;}
        }
}

likelihood(x,L)
float *x;
double *L;
{
  double lik,Ht,z[10];
  int i,j;

  lik=0;
  for (i=P-1;i<SIZE;i++)
  {
  z[0]=1;
  for(j=1;j<P;j++) z[j]=yhl_yt[i-j]*yhl_yt[i-j];
  Ht=0;
  for(j=1;j<=P;j++) Ht+=x[j]*z[j-1];
```

```
        if(Ht<.001) Ht=.001;
    lik +=(-.5)*log(Ht)-.5/Ht*yhl_yt[i]*yhl_yt[i];
  }
  (*L)=lik;

}
#define FREERETURN {free_matrix(alpha,1,n,1,n);free_vector(bet,1,n);\
        free_ivector(indx,1,n);return;}

mnewt(ntrial,x,n,tolx,tolf)
int ntrial,n;
float x[],tolx,tolf;
{
        int k,i,*indx,*ivector();
        float errx,errf,d,*bet,**alpha,*vector(),**matrix();
        void usrfun(),ludcmp(),lubksb(),free_ivector(),free_vector(),
                free_matrix();

        indx=ivector(1,n);
        bet=vector(1,n);
        alpha=matrix(1,n,1,n);
        for (k=1;k<=ntrial;k++) {
                usrfun(x,alpha,bet);
        printf("a1=%f a2=%f %f %f\n",x[1],x[2],bet[1],bet[2]);
                errf=0.0;
                for (i=1;i<=n;i++) errf += fabs(bet[i]);
                if (errf <= tolf) FREERETURN
                ludcmp(alpha,n,indx,&d);
                lubksb(alpha,n,indx,bet);
                errx=0.0;
                for (i=1;i<=n;i++) {
                        errx += fabs(bet[i]);
                        x[i] += bet[i];
                }
                if (errx <= tolx) FREERETURN
        }
        FREERETURN
}

#undef FREERETURN
void lubksb(a,n,indx,b)
float **a,b[];
int n,*indx;
{
        int i,ii=0,ip,j;
        float sum;

        for (i=1;i<=n;i++) {
                ip=indx[i];
                sum=b[ip];
                b[ip]=b[i];
                if (ii)
                        for (j=ii;j<=i-1;j++) sum -= a[i][j]*b[j];
                else if (sum) ii=i;
```

```
		b[i]=sum;
	}
	for (i=n;i>=1;i--) {
		sum=b[i];
		for (j=i+1;j<=n;j++) sum -= a[i][j]*b[j];
		b[i]=sum/a[i][i];
	}
}

#define TINY 1.0e-20;

void ludcmp(a,n,indx,d)
int n,*indx;
float **a,*d;
{
	int i,imax,j,k;
	float big,dum,sum,temp;
	float *vv,*vector();
	void nrerror(),free_vector();

	vv=vector(1,n);
	*d=1.0;
	for (i=1;i<=n;i++) {
		big=0.0;
		for (j=1;j<=n;j++)
			if ((temp=fabs(a[i][j])) > big) big=temp;
		if (big == 0.0) nrerror("Singular matrix in routine LUDCMP");
		vv[i]=1.0/big;
	}
	for (j=1;j<=n;j++) {
		for (i=1;i<j;i++) {
			sum=a[i][j];
			for (k=1;k<i;k++) sum -= a[i][k]*a[k][j];
			a[i][j]=sum;
		}
		big=0.0;
		for (i=j;i<=n;i++) {
			sum=a[i][j];
			for (k=1;k<j;k++)
				sum -= a[i][k]*a[k][j];
			a[i][j]=sum;
			if ( (dum=vv[i]*fabs(sum)) >= big) {
				big=dum;
				imax=i;
			}
		}
		if (j != imax) {
			for (k=1;k<=n;k++) {
				dum=a[imax][k];
				a[imax][k]=a[j][k];
				a[j][k]=dum;
			}
			*d = -(*d);
			vv[imax]=vv[j];
```

```
			}
			indx[j]=imax;
			if (a[j][j] == 0.0) a[j][j]=TINY;
			if (j != n) {
				dum=1.0/(a[j][j]);
				for (i=j+1;i<=n;i++) a[i][j] *= dum;
			}
		}
		free_vector(vv,1,n);
}

#undef TINY

void nrerror(error_text)
char error_text[];
{
		void exit();

		fprintf(stderr,"Numerical Recipes run-time error...\n");
		fprintf(stderr,"%s\n",error_text);
		fprintf(stderr,"...now exiting to system...\n");
		exit(1);
}

float *vector(nl,nh)
int nl,nh;
{
		float *v;

		v=(float *)malloc((unsigned) (nh-nl+1)*sizeof(float));
		if (!v) nrerror("allocation failure in vector()");
		return v-nl;
}

int *ivector(nl,nh)
int nl,nh;
{
		int *v;

		v=(int *)malloc((unsigned) (nh-nl+1)*sizeof(int));
		if (!v) nrerror("allocation failure in ivector()");
		return v-nl;
}

double *dvector(nl,nh)
int nl,nh;
{
		double *v;

		v=(double *)malloc((unsigned) (nh-nl+1)*sizeof(double));
		if (!v) nrerror("allocation failure in dvector()");
		return v-nl;
}
```

```
float **matrix(nrl,nrh,ncl,nch)
int nrl,nrh,ncl,nch;
{
	int i;
	float **m;

	m=(float **) malloc((unsigned) (nrh-nrl+1)*sizeof(float*));
	if (!m) nrerror("allocation failure 1 in matrix()");
	m -= nrl;

	for(i=nrl;i<=nrh;i++) {
		m[i]=(float *) malloc((unsigned) (nch-ncl+1)*sizeof(float));
		if (!m[i]) nrerror("allocation failure 2 in matrix()");
		m[i] -= ncl;
	}
	return m;
}

double **dmatrix(nrl,nrh,ncl,nch)
int nrl,nrh,ncl,nch;
{
	int i;
	double **m;

	m=(double **) malloc((unsigned) (nrh-nrl+1)*sizeof(double*));
	if (!m) nrerror("allocation failure 1 in dmatrix()");
	m -= nrl;

	for(i=nrl;i<=nrh;i++) {
		m[i]=(double *) malloc((unsigned) (nch-ncl+1)*sizeof(double));
		if (!m[i]) nrerror("allocation failure 2 in dmatrix()");
		m[i] -= ncl;
	}
	return m;
}

int **imatrix(nrl,nrh,ncl,nch)
int nrl,nrh,ncl,nch;
{
	int i,**m;

	m=(int **)malloc((unsigned) (nrh-nrl+1)*sizeof(int*));
	if (!m) nrerror("allocation failure 1 in imatrix()");
	m -= nrl;

	for(i=nrl;i<=nrh;i++) {
		m[i]=(int *)malloc((unsigned) (nch-ncl+1)*sizeof(int));
		if (!m[i]) nrerror("allocation failure 2 in imatrix()");
		m[i] -= ncl;
	}
	return m;
}

float **submatrix(a,oldrl,oldrh,oldcl,oldch,newrl,newcl)
```

```
float **a;
int oldrl,oldrh,oldcl,oldch,newrl,newcl;
{
	int i,j;
	float **m;

	m=(float **) malloc((unsigned) (oldrh-oldrl+1)*sizeof(float*));
	if (!m) nrerror("allocation failure in submatrix()");
	m -= newrl;

	for(i=oldrl,j=newrl;i<=oldrh;i++,j++) m[j]=a[i]+oldcl-newcl;

	return m;
}

void free_vector(v,nl,nh)
float *v;
int nl,nh;
{
	free((char*) (v+nl));
}

void free_ivector(v,nl,nh)
int *v,nl,nh;
{
	free((char*) (v+nl));
}

void free_dvector(v,nl,nh)
double *v;
int nl,nh;
{
	free((char*) (v+nl));
}

void free_matrix(m,nrl,nrh,ncl,nch)
float **m;
int nrl,nrh,ncl,nch;
{
	int i;

	for(i=nrh;i>=nrl;i--) free((char*) (m[i]+ncl));
	free((char*) (m+nrl));
}

void free_dmatrix(m,nrl,nrh,ncl,nch)
double **m;
int nrl,nrh,ncl,nch;
{
	int i;

	for(i=nrh;i>=nrl;i--) free((char*) (m[i]+ncl));
	free((char*) (m+nrl));
}
```

```
void free_imatrix(m,nrl,nrh,ncl,nch)
int **m;
int nrl,nrh,ncl,nch;
{
	int i;

	for(i=nrh;i>=nrl;i--) free((char*) (m[i]+ncl));
	free((char*) (m+nrl));
}

void free_submatrix(b,nrl,nrh,ncl,nch)
float **b;
int nrl,nrh,ncl,nch;
{
	free((char*) (b+nrl));
}

float **convert_matrix(a,nrl,nrh,ncl,nch)
float *a;
int nrl,nrh,ncl,nch;
{
	int i,j,nrow,ncol;
	float **m;

	nrow=nrh-nrl+1;
	ncol=nch-ncl+1;
	m = (float **) malloc((unsigned) (nrow)*sizeof(float*));
	if (!m) nrerror("allocation failure in convert_matrix()");
	m -= nrl;
	for(i=0,j=nrl;i<=nrow-1;i++,j++) m[j]=a+ncol*i-ncl;
	return m;
}


void free_convert_matrix(b,nrl,nrh,ncl,nch)
float **b;
int nrl,nrh,ncl,nch;
{
	free((char*) (b+nrl));
}
```

References

Akaike, H. (1973), "Information Theory and An Extension of the Maximum Likelihood Principle." *2nd International Symposium on Information Theory*, B. N., Petrov and F. Csaki (eds.), Akademiai Kiado, Budapest, 2678-281.

Black, Fischer and Myron Scholes (1973), "The Pricing of Options and Corporate Liabilities." *Journal of Political Economy*, **81**, 637-659.

Bollerslev, T. (1986), "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics*, **31**, 307-328.

Box, G. E. P. and G. M. Jenkins (1970), *Time Series Analysis: Forecasting and Control*, Holden-Day.

Brockwell, Peter J. and Richard A. Davis (1987), *Time Series: Theory and Methods*, Springer-Verlag.

Clark, P. K. (1973), "A Subordinate Stochastic Process Model With Finite Variance for Speculative Price." *Econometrica*, **41**, 135-155.

Cox, John C. and Mark Rubinstein, (1985), *Options Markets*, Prentice-Hall.

Engle, Robert F. (1980), "Estimates of the Variance of U.S. Inflation Based on the ARCH Model." University of California, San Diego Discussion Paper 80-14.

Engle, Robert F. (1982), "Autoregressive Conditional Heteroscedasticity With Estimates of the Variance of United Kingdom Inflation." *Econometrica*, **50**, 987-1008.

Hoque, Tareq I. (1992), "Second-moment Market Efficiency: Testing the Volatility of Foreign Exchange Prices." , Master's Thesis, MIT Sloan School of Management.

Tauchen, G. E. and M. Pitts (1983), "The Price Variability-Volume Relationship on Speculative Markets." *Econometrica*, **51**, 485-505.

Taylor, Stephen J. (1986), *Modelling Financial Time Series*, John Wiley & Sons.

Zhou, Bin (1992), "High Frequency Data and Volatility in Foreign Exchange Rates." MIT Sloan School Working Paper 3485-92.