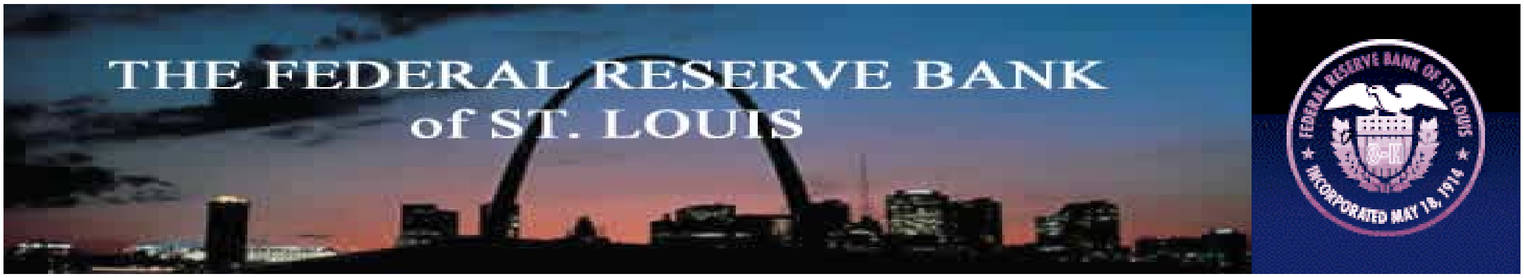


WORKING PAPER SERIES

Predicting Exchange Rate Volatility: Genetic Programming vs. GARCH and RiskMetrics™

Christopher J. Neely
and
Paul A. Weller

Working Paper 2001-009B
http://research.stlouisfed.org/wp/2001/2001-009.pdf

FEDERAL RESERVE BANK OF ST. LOUIS
Research Division
411 Locust Street
St. Louis, MO 63102

Predicting Exchange Rate Volatility: Genetic Programming vs. GARCH and RiskMetrics™

Christopher J. Neely
Research Department
Federal Reserve Bank of St. Louis
P.O. Box 442
St. Louis, MO 63166
USA
neely@stls.frb.org

Paul A. Weller
Department of Finance
Henry B. Tippie College of Business Administration
University of Iowa
Iowa City, IA 52242
USA
paul-weller@uiowa.edu

Revised: September 21, 2001

Abstract: This article investigates the use of genetic programming to forecast out-of-sample daily volatility in the foreign exchange market. Forecasting performance is evaluated relative to GARCH(1,1) and RiskMetrics models for two currencies, DEM and JPY. Although the GARCH/RiskMetrics models appear to have a inconsistent marginal edge over the genetic program using the mean-squared-error (MSE) and R^2 criteria, the genetic program consistently produces lower mean absolute forecast errors (MAE) at all horizons and for both currencies.

Keywords genetic programming, GARCH, foreign exchange, volatility, forecasting, heteroskedasticity.

Acknowledgments: This paper is a revised and expanded version of a chapter entitled "Using A Genetic Program To Predict Exchange Rate Volatility," in Genetic Programming in Computational Finance, edited by Shu-Heng Chen, published by Kluwer. We would like to thank Janis Zvingelis for excellent research assistance. The views expressed are those of the authors and do not necessarily represent the views of the Federal Reserve Bank of St. Louis, the Federal Reserve System or the Board of Governors.

Introduction

It is well-established that the volatility of asset prices displays considerable persistence. That is, large movements in prices tend to be followed by more large moves, producing positive serial correlation in squared returns. Thus current and past volatility can be used to predict future volatility. This fact is important to both financial market practitioners and regulators.

Professional traders in equity and foreign exchange markets must pay attention not only to the expected return from their trading activity but also to the risk that they incur. Risk averse investors will wish to reduce their exposure during periods of high volatility and improvements in risk-adjusted performance depend upon the accuracy of volatility predictions. Many current models of risk management, such as Value-at-Risk (VaR), use volatility predictions as inputs.

The bank capital adequacy standards recently proposed by the Basle Committee on banking supervision illustrate the importance of sophisticated risk management techniques for regulators. These norms are aimed at providing international banks with greater incentives to manage financial risk in a sophisticated fashion, so that they might economize on capital. One such system that is widely used is RiskMetrics, developed by J.P. Morgan.

A core component of the RiskMetrics system is a statistical model—a member of the large ARCH/GARCH family—that forecasts volatility. Such ARCH/GARCH models are parametric. That is, they make specific assumptions about the functional form of the data generation process and the distribution of error terms. Parametric models such as GARCH are easy to estimate and readily interpretable, but these advantages may come at a cost. Other, perhaps much more complex models may be better representations of the underlying data generation process. If so, then procedures designed to identify these alternative models have an obvious payoff. Such procedures are described as non-parametric. Instead of specifying a particular functional form for the data generation process and making distributional assumptions about the error terms, a non-parametric procedure will search for the best fit over a large set of alternative functional forms.

This article investigates the performance of a genetic program applied to the problem of forecasting volatility in the foreign exchange market. Genetic programming is a computer search and problem-solving methodology that can be adapted for use in non-parametric estimation. It has been shown to detect patterns in the conditional mean of foreign exchange and equity returns that are not accounted for by standard statistical models (Neely, Weller, and Dittmar 1997; Neely and Weller 1999 and 2001; Neely 2000). This suggests that a genetic program may also be a powerful tool for generating predictions of asset price volatility.

We compare the performance of a genetic program in forecasting daily exchange rate volatility for the dollar-deutschemark and dollar-yen exchange rates with that of a GARCH(1, 1) model and a related RiskMetrics volatility forecast (described in the following section). These models are widely used both by academics and practitioners and thus are good benchmarks to which to compare the genetic program forecasts. While the overall forecast performance of the two methods is broadly similar, on some dimensions the genetic program produces significantly superior results. This is an encouraging finding, and suggests that more detailed investigation of this methodology applied to volatility forecasting would be warranted.

1. The Benchmark Model

Before discussing the genetic programming procedure, we will review the benchmark GARCH and RiskMetrics volatility models. Engle (1982) developed the Autoregressive

Conditionally Heteroskedastic (ARCH) model to characterize the observed serial correlation in asset price volatility. Suppose we assume that a price P_t follows a random walk

$$P_{t+1} = P_t + \varepsilon_{t+1} \tag{1}$$

where $\varepsilon_{t+1} \sim N(0, \sigma_t^2)$. The variance of the error term depends upon *t*, and the objective of the model is to characterize the way in which this variance changes over time. The ARCH model assumes that this dependence can be captured by an autoregressive process of the form

$$\sigma_t^2 = \omega + \alpha_0 \varepsilon_t^2 + \alpha_1 \varepsilon_{t-1}^2 + \cdots + \alpha_m \varepsilon_{t-m}^2 \tag{2}$$

where the restrictions $\omega \geq 0$, $\alpha_i \geq 0$ for i = 0, 1, …, m ensure that the predicted variance is always nonnegative. This specification illustrates clearly how current levels of volatility will be influenced by the past, and how periods of high or low price fluctuation will tend to persist.

Bollerslev (1986) extended the ARCH class to produce the Generalized Autoregressive Conditionally Heteroskedastic (GARCH) model, in which the variance is given by

$$\sigma_t^2 = \omega + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2 + \cdots \beta_k \sigma_{t-k}^2 + \alpha_0 \varepsilon_t^2 + \alpha_1 \varepsilon_{t-1}^2 + \cdots + \alpha_m \varepsilon_{t-m}^2 . \tag{3}$$

The simplest specification in this class, and the one most widely used, is referred to as GARCH(1, 1) and is given by

$$\sigma_t^2 = \omega + \beta \sigma_{t-1}^2 + \alpha \varepsilon_t^2 \tag{4}$$

When $\alpha + \beta < 1$, the variance process displays mean reversion to the unconditional expectation of σ_t^2, $\omega / (1 - \alpha - \beta)$. That is, forecasts of volatility in the distant future will be equal to the unconditional expectation of σ_t^2, $\omega / (1 - \alpha - \beta)$.

The RiskMetrics model for volatility forecasting imposes the restrictions that $\alpha + \beta = 1$ and that $\omega = 0$.[1] In addition, the parameter β is not estimated, but imposed to be equal to 0.94 (J.P. Morgan, 1996). This value was found to minimize the mean squared error of volatility forecasts for asset prices. The RiskMetrics one-day ahead volatility forecast is

$$\sigma_t^2 = \beta\sigma_{t-1}^2 + (1-\beta)\varepsilon_t^2 \tag{5}$$

The GARCH model has been used to characterize patterns of volatility in U.S. dollar foreign exchange markets (Baillie and Bollerslev 1989 and 1991) and in the European Monetary System (Neely (1999)). However, initial investigations into the explanatory power of out-of-sample forecasts produced disappointing results (West and Cho, 1995). Jorion (1995) found that volatility forecasts for several major currencies from the GARCH model were outperformed by implied volatilities generated from the Black-Scholes option-pricing model. These studies typically used the squared daily return as the variable to be forecast. However, this is a very imprecise measure of true, unobserved volatility. For example, the exchange rate may move around a lot during the day, and yet end up close to its value the same time the previous day. The squared daily return will be small, even though volatility was high. More recently, it has been demonstrated that one can significantly improve the forecasting power of the GARCH model by measuring volatility as the sum of intraday squared returns (Andersen and Bollerslev, 1998). This measure is referred to as integrated, or realized volatility. In theory, if the true underlying price path is a diffusion process it is possible to obtain progressively more accurate estimates of the true volatility by increasing the frequency of intraday observation. Of course, there are practical limits to this; microstructural effects begin to degrade accuracy beyond a certain point.

[1] The restriction, $\alpha + \beta = 1$, implies that shocks to the volatility process persist forever; higher volatility today will lead one to forecast higher volatility indefinitely. It therefore falls into the class of integrated GARCH, or IGARCH models.

2. Genetic Algorithms and Genetic Programming

Genetic algorithms are computer search procedures used to solve appropriately defined problems. The structure of the search procedure is based on the principles of natural selection. These procedures were developed for genetic algorithms by Holland (1975) and extended to *genetic programming* by Koza (1992). The essential features of both algorithms include (1) a means of representing potential solutions to a problem as character strings which can be split up and recombined to form new potential solutions and (2) a fitness criterion which measures the "quality" of a candidate solution. Both types of algorithms produce successive "generations" of candidate solutions using procedures which mimic genetic reproduction and recombination. Each new generation is subjected to the pressures of "natural selection" by increasing the probability that candidate solutions scoring highly on the fitness criterion get to reproduce.

To understand the principles involved in genetic programming, it is useful to understand the operation of the simpler genetic algorithm. Genetic algorithms require that potential solutions be expressed as fixed length character strings. Consider a problem in which candidate solutions are mapped into binary strings s of length five. One possible solution would be represented as (01010). Associated with this binary string would be a measure of fitness that quantifies how well it solves the problem. In other words, we need a fitness function $m(s)$ that maps the strings into the real line and thus ranks the quality of the solutions. Next we introduce the *crossover operator*. Given two strings, a crossover point is randomly selected and the first part of one string is combined with the second part of the other. For example, given the two strings (00101) and (11010) and a crossover point between elements two and three, the new string (00010) is generated. The string consisting of the remaining parts of the original strings is discarded.

The algorithm begins by randomly generating an *initial population* of binary strings and then evaluates the fitness of each string by applying the fitness function $m(s)$. Next the program produces a new (second) generation of candidate solutions by selecting pairs of strings at random from this initial population and applying the crossover operator to create new strings. The probability of selecting a given string is set to be proportional to its fitness. Thus a "selection pressure" in favor of progressively superior solutions is introduced. This process is repeated to produce successive generations of strings, keeping the size of each generation the same. The procedure "evolves" new generations of improved potential solutions.

Recall that genetic algorithms require that potential solutions be encoded as fixed length character strings. Koza's (1992) extension, genetic programming, instead employs variable-length, hierarchical strings that can be thought of as decision trees or computer programs. However, the basic structure of a genetic program is exactly the same as described above. In particular, the crossover operator is applied to pairs of decision trees to generate new "offspring" trees.

The application in this paper represents forecasting functions as trees, and makes use of the following function set in constructing them: plus, minus, times, divide, norm, log, exponential, square root, and cumulative standard normal distribution function. In addition, we supply the following set of data functions: *data, average, max, min,* and *lag*. The data functions can operate on any of the four data series we permit as inputs to the genetic program: (1) daily foreign exchange returns; (2) integrated volatility (i.e., the sum of squared intraday returns); (3) the sum of the absolute value of intraday returns; and (4) number of days until the next business day. For example, *data*(*returns*(t)) is simply the identity function which computes the daily return at t. The other data functions operate in a similar fashion, but also take numerical arguments to specify the length of the window—the number of observations—over which the functions operate. The numerical arguments that the functions take are determined by the genetic program. Thus *average*(*returns*(t))(n) generates the arithmetic average of the return observations $t, t-1, \ldots, t-n+1$.

The choice of elements to include in the function set is a potentially important one. While a genetic program can, in principle, produce a very highly complex solution from simple functions, computational limitations might make such solutions very difficult to find in practice. Providing specialized functions to the genetic program that are thought to be useful to a "good" solution to the problem can greatly increase the efficiency of the search by encouraging the genetic program to search in the area of the solution space containing those functions. On the other hand, this might bias the genetic program's search away from other promising regions. To focus search in promising regions of the solution space, we investigate the results of adding three more complex data functions described below to the original set of functions.

The expanded set of data functions consists of the original set plus *geo, mem,* and *arch5*. Each of these functions approximates the forecast of a known parametric model of conditional volatility. Thus, the genetic program might find them useful. The function *geo* returns the following weighted average of 10 lags of past data:

$$geo(data)(\alpha) \equiv \sum_{j=0}^{9} \alpha\left[(1-\alpha)^j\right] data_{t-j} \ . \tag{6}$$

This function can be derived from the prediction of an IGARCH specification with parameter α, where we constrain α to satisfy $0.01 \le \alpha \le 0.99$ and lags are truncated at 10. The function *mem* returns a weighted sum similar to that which would be obtained from a long memory specification for volatility. It takes the form

$$mem(data)(d) \equiv \sum_{j=0}^{9} h_j data_{t-j} \tag{7}$$

where $h_0 = 1$, $h_j \propto (1/j!)(d+j-1)(d+j-2)...(d+1)d$ for $j > 0$ and the sum of the coefficients h_j is constrained to equal one so that the output would be of the same magnitude as recent volatility. The parameter d is determined by the genetic program and constrained to satisfy $-1 < d < 1$. Finally, the function *arch5* permits a flexible weighting of the five most recent observations, where the values for h_j are provided by the genetic program and constrained to lie within {-5,5} and to sum to one. Again the constraint on the sum of the coefficients ensures that the magnitude of the output will be similar to that of recent volatility. The function has the form

$$arch5(data)(h) \equiv \sum_{j=0}^{4} h_j data_{t-j}, \quad h = (h_0, h_1, \ldots, h_4) \tag{8}$$

Figure 1 illustrates a simple example of a hypothetical tree determining a forecasting function. The function first computes the maximum of the sum of squared intraday returns over the last five days. This number is multiplied by 0.1 and the result entered as the argument x of the function $(8/\pi)\arctan(x)+4$. This latter function is common to all trees and maps the real line into the interval (0, 8). It ensures that all forecasts are nonnegative and bounded above by a number chosen with reference to the characteristics of the in-sample period.

[Place Figure 1 about here]

We now turn to the form of the fitness criterion. Because true volatility is not directly observed, it is necessary to use an appropriate proxy in order to assess the volatility forecasting performance of the genetic program. One possibility is to use the ex post squared daily return. However, as Andersen and Bollerslev (1998) have pointed out, this is an extremely noisy measure of the true underlying volatility, and is largely responsible for the apparently poor forecast performance of GARCH models. A better approach is to sum intraday returns to measure true daily volatility (i.e., integrated volatility) more accurately. We measure integrated volatility using five irregularly spaced intraday observations. If $S_{i,t}$ is the i-th observation on date *t*, we define

$$\begin{aligned} R_{i,t} &= 100 \cdot \ln\left(\frac{S_{i+1,t}}{S_{i,t}}\right) \qquad \text{for i} = 1, 2, 3, \text{and } 4 \\ &= 100 \cdot \ln\left(\frac{S_{1,t+1}}{S_{5,t}}\right) \qquad \text{for i} = 5 \end{aligned} \tag{9}$$

$$\sigma_{I,t}^2 = \sum_{i=1}^{5} R_{i,t}^2 \,. \tag{10}$$

Thus $\sigma_{I,t}^2$ is the measure of integrated volatility on date *t*.[2] Using five intraday observations represents a compromise between the increase in accuracy generated by more frequent observations and the problems of data handling and availability that arise as one moves to progressively higher frequencies of intraday observation.

In constructing the rules, the genetic program minimized the mean square forecast error (MSE) as the fitness criterion. There are potential inefficiencies involved in using this criterion on heteroskedastic data. However, a heteroskedasticity-corrected fitness measure proved unsatisfactory in experiments. With three to five observations per day, there were instances

[2] More precisely, daily volatility is calculated from 1700 GMT to 1700 GMT.

where the integrated daily volatility was very small; the heteroskedasticity correction caused the measure to be inappropriately sensitive to those observations.[3]

3. Data and Implementation

The object of this exercise is to forecast the daily volatility (the sum of intraday squared returns) of two currencies against the dollar, the German mark (DEM) and Japanese yen (JPY), over the period June 1975 to September 1999. Thus, the final nine months of data for the DEM represent the rate derived from that of the euro, which superseded the DEM in January 1999. The timing of observations was 1000, 1400, 1600, 1700, and 2200 GMT. Days with fewer than three valid observations or no observation at 1700 were treated as missing. In addition, weekends were excluded. The sources of the data for both exchange rates are summarized in Table 1. We provided the genetic program with three series in addition to the integrated volatility series,: daily returns, sum of absolute intraday returns, and the number of days until the next trading day.

[Place Table 1 about here]

The full sample is divided into three subperiods: the training period June 1975 – December 1979; the selection period January 1980 – December 30, 1986; and the out-of-sample period December 31, 1986 – September 21, 1999. The role of these subperiods is described below.

---

[3] A perennial problem with using flexible, powerful search procedures like genetic programming is overfitting—the finding of spurious patterns in the data. Given the well-documented tendency for the genetic program to overfit the data it is necessary to design procedures to mitigate this (e.g., Neely, Weller and Dittmar (1997)). Here, we investigated the effect of modifying the fitness criterion by adding a penalty for complexity. This consisted of subtracting an amount (0.002 * number of nodes) from the negative MSE. Nodes are data and numerical functions. This modification is intended to bias the search toward functions with fewer nodes, which are simpler and therefore

In searching through the solution space of forecasting functions, the genetic program followed the procedures below.

1. Create an initial generation of 500 randomly generated forecast functions.
2. Measure the MSE of each function over the training period and rank according to performance.
3. Select the function with the lowest MSE and calculate its MSE over the selection period. Save it as the initial best forecast function.
4. Select two functions at random, using weights attaching higher probability to more highly-ranked functions. Apply the crossover operator to create a new function, which then replaces an old function, chosen using weights attaching higher probability to less highly-ranked functions. Repeat this procedure 500 times to create a new generation of functions.
5. Measure the MSE of each function in the new generation over the training period. Take the best function in the training period and evaluate the MSE over the selection period. If it outperforms the previous best forecast, save it as the new best forecast function.
6. Stop if no new best function appears for 25 generations, or after 50 generations. Otherwise, return to stage 4.

The stages above describe one trial. Each trial produces one forecast function. The results of each trial will generally differ as a result of sampling variation. For this reason it is necessary to run a number of trials and then to aggregate the results. The aggregation methods are described in the following section.

less prone to overfit the data. Unfortunately, this procedure produced no significant changes in performance, so we will only report results from the unmodified version.

4. Results

The benchmark results are those from the GARCH(1, 1) and RiskMetrics models described in Section 2, estimated over the in-sample period June 1975 to December 30, 1986. We generate daily integrated volatility (defined in equations (9) and (10)) forecasts from these models, in and out of sample, at horizons of one, five, and twenty days.[4]

We also forecast with a genetic program whose training and selection periods coincide with the in-sample estimation period for the GARCH model. For each case of the genetic program we generated ten trials, each of which produced a forecast function. The cases were distinguished by the following factors: (1) forecast horizon – one, five and twenty days; (2) the number of data functions – five or eight. For each case, we generated ten rules. The forecasts from each set of ten rules were aggregated in two ways. The equally weighted forecast is the arithmetic average of the forecasts from each of the ten trials. The median-weighted forecast takes the median forecast from the set of ten forecasts at each date. We report six measures of out-of-sample forecast performance: mean square error, mean absolute error (MAE), R^2, mean forecast bias, kernel estimates of the error densities and generalized mean square forecast error matrix tests.

[Place Figure 2 about here]

Before discussing the results, we first present a simple example of the forecasting rules produced by the genetic program. Figure 2 illustrates a one-day-ahead forecasting function for the DEM. Its out-of-sample MSE was 0.496. The function is interpreted as follows. The number

[4] Note that the forecasted variable at the five-day (twenty-day) horizon is the integrated volatility five (twenty) days in the future. It is not the sum of the next five (twenty) days of integrated volatility.

– 0.4744 at the terminal node enters as the argument of *geo*(sum of squared intraday returns). Since the argument of *geo* is constrained to lie between 0.01 and 0.99, it is set to 0.01. The number generated by this function then enters as the argument in *geo*(Ndays) where Ndays refers to the data series "number of days to the next trading day". We caution that this example was chosen largely because of its relatively simple form; some trials generated considerably more complex rules with as many as 10 levels and/or 100 nodes.

Table 2 reports in-sample results for the baseline case with five data functions. The figures for MSE for the DEM are very similar for the GARCH and equally weighted genetic program forecasts at the 1- and 5-day horizons, but the genetic program is appreciably better at the 20-day horizon. The median weighted forecast is generally somewhat inferior to the equally weighted forecast, but follows the same pattern over the forecast horizons relative to the GARCH model. That is, its best relative performance is at the twenty-day horizon. The RiskMetrics forecasts also are generally comparable to GARCH forecasts at 1- and 5-day horizons, but a bit better at longer horizons. For the JPY, the genetic program produces equally weighted MSE figures which are in all cases lower than for the GARCH and RiskMetrics models. Similarly, the equally weighted genetic programming rules have higher R^2s over each horizon than the GARCH and RiskMetrics models. This is not especially surprising given the flexibility of the non-parametric procedure and its known tendency to overfit in sample.

[Place Table 2 about here]

Table 3 presents a more interesting comparison, out-of-sample performance over the period December 31, 1986 to September 21, 1999. The equally weighted genetic program MSE figures are usually slightly larger than those of the GARCH and RiskMetrics forecasts at all horizons for both currencies. The genetic programming R^2s are similarly typically slightly smaller than those of the GARCH/RiskMetrics forecasts. However, the equally weighted genetic program has lower MAE than do the GARCH/RiskMetrics models at all horizons and for both currencies.

[Place Table 3 about here]

Table 4 reports the out-of-sample performance of the genetic program forecasts using the augmented set of data functions, which include *geo*, *mem,* and *arch5*. For ease of comparison Table 4 repeats the out-of-sample figures for the GARCH model. The MSE and R^2 statistics from this table are more equivocal than those from Table 3. The equally weighted genetic program MSE for the DEM cases are slightly larger than those of the GARCH and RiskMetrics forecasts at the one- and five-day horizons, but the genetic program performs rather better than GARCH at the twenty-day horizon. This is not, however, reflected in the R^2, for which the GARCH/RiskMetrics models are better at longer horizons. For the JPY the situation is reversed, the equally weighted genetic programming MSE is lower than the GARCH/RiskMetrics figures at the one-day horizon but larger at the 20-day horizon. The equally weighted genetic program also has a slight edge in R^2 at the 1-day horizon. The figures for the MAE of the genetic program are very little changed from Table 3, and are still substantially better than those of the GARCH/RiskMetrics predictions.

[Place Table 4 about here]

To summarize: with MSE as the performance criterion, neither the genetic programming procedure nor the GARCH/RiskMetrics model is clearly superior. The GARCH/RiskMetrics models do achieve slightly higher R^2s at longer horizons but the MAE criterion clearly prefers the genetic programming forecasts. In both tables, there is some tendency for the median weighted genetic programming forecast to perform less well than the equally weighted. The out-of-sample RiskMetrics forecasts are usually marginally better than those of the estimated GARCH model by MSE and MAE criteria but marginally worse when judged by R^2.

Comparing the genetic programming results in Table 4 to Table 3 shows that expanding the set of data functions leads to only a marginal improvement in the performance of the genetic program. Therefore further results will concentrate on out-of-sample forecasts in the baseline genetic programming case presented in Table 3, where only five data functions were used. We present kernel estimates of the densities of out-of-sample forecast errors at the various horizons in Figures 3 to 5.[5]

[Place Figures 3 to 5 about here]

The most striking feature to emerge from these figures is the apparent bias in the GARCH forecasts when compared to their genetic program counterparts. At all forecast horizons and for both currencies there is a positive shift in the error distributions of the GARCH forecasts that move the modes of the forecast densities away from zero. However, the relative magnitude of the bias in the mode does not carry over to the mean. Table 5 shows that though both forecasts

[5] We choose to graph the density of the GARCH errors as the density of the RiskMetrics errors will have a mean of approximately zero by construction.

are biased in the mean, the magnitude of the bias is considerably greater for the genetic program. Tests for the bias—carried out with a Newey-West correction for serial correlation—show that all the forecasts are biased in a statistically significant way (Newey and West 1987 and 1994). The evidence from Figures 3 to 5—that the modes of the genetic programming error distribution are closer to zero than those of the GARCH model—indicates that the bias in the genetic programming forecasts is being substantially influenced by a small number of negative outliers.

[Place Table 5 about here]

The MSE and R^2 evidence presented so far fails to indicate a clear preference for any of the 4 sets of forecasts. The best model varies by forecast horizon and by forecast evaluation criterion. This confused state of affairs leaves one wondering whether these disparate results can be reconciled to produce an unambiguous ranking of the two methodologies. One method by which multi-horizon forecasts from 2 sources can be aggregated and compared is the generalized forecast error second moment (GFESM) method proposed by Clements and Hendry (1993). Unfortunately, this method has some drawbacks. For example, the GFESM can prefer model 1 to model 2 based on forecasts from horizon 1 to horizon h, even if model 2's forecasts dominate at every forecast horizon up to h. To remedy the perceived weaknesses in the GFESM, Newbold, Harvey and Leybourne (1999) proposed the generalized mean squared forecast error matrix (GMSFEM) criterion. This procedure prefers forecasting method 1 to method 2 if the magnitude of all linear combinations of forecast errors is at least as small under method 1 as method 2.

To explain the GMSFEM more fully, let us introduce some notation. The 1-by-3 vector of 1-, 5- and 20-day GARCH forecast errors at time t is $e_t^{GARCH} = \{e_{t,1}^{GARCH}, e_{t,5}^{GARCH}, e_{t,20}^{GARCH}\}$, and the second moment matrix of these forecast errors is $\Phi^{GARCH} = E\left(e_t^{GARCH} e_t^{GARCH}{}'\right)$. The RiskMetrics and genetic programming variables are defined analogously. The GMSFEM says that the GARCH model is preferred to the genetic programming forecasts if every linear combination of GARCH forecast errors is at least as small as every linear combination of genetic programming forecast errors. That is, if

$$d'\left(\Phi^{GARCH} - \Phi^{GP}\right)d \leq 0 \qquad \text{for all d} \neq 0 \qquad .^{6} \tag{11}$$

This condition is met if every eigenvalue of the matrix $\left(\Phi^{GARCH} - \Phi^{GP}\right)$ is nonpositive and at least one is negative. Clearly, the criterion prefers the genetic programming forecast if every eigenvalue is nonnegative and at least one is positive.

[Place Table 6 about here]

Table 6 shows four sets of eigenvalues from the $\left(\Phi^{GARCH} - \Phi^{GP}\right)$ matrix, using both the equally weighted and median weighted genetic program forecasts, for both exchange rates. It confirms the previous results. The only case in which there are all negative (or positive) eigenvalues is the comparison of the RiskMetrics forecast to the median weighted genetic programming forecast. In that case, all the eigenvalues are negative, indicating that the RiskMetrics forecasts dominate the median weighted genetic programming forecasts under the GMSFEM criterion. In every other set of eigenvalues there are both positive and negative

[6] Note that the GMSFEM criterion implicitly favors models that do well in terms of MSE, rather than in terms of MAE.

values. Neither GARCH/RiskMetrics forecasts nor genetic programming forecasts dominate the other under the GMSFEM criterion.

5. Discussion and Conclusion

We choose to use the problem of forecasting conditional volatility in the foreign exchange market to illustrate the strengths and weaknesses of genetic programming because it is a challenging problem with a well accepted benchmark solution, the GARCH(1,1) model. The genetic program did reasonably well in forecasting out-of-sample volatility. While the genetic programming rules did not usually match the GARCH(1,1) or RiskMetrics models' MSE or R^2, its performance on those measures was generally close. But the genetic program did consistently outperform the GARCH model on mean absolute error (MAE) and modal error bias at all horizons. The genetic programming solutions appeared to suffer from some in-sample overfitting, which was not mitigated, in this case, by an ad hoc penalty for rule complexity.

Our results suggest some interesting issues for further investigation. The superiority of the genetic program according to the MAE criterion is perhaps surprising given that we used MSE as the fitness criterion. This raises the possibility that further improvement in the forecasting performance of the genetic program relative to the GARCH model could be achieved by using MAE as the fitness criterion. Also, given that increasing the frequency of intraday observations has been shown to improve the accuracy of forecasts based on the GARCH model (Andersen et al., 2001), it is important to know whether the results of this investigation survive in that context.

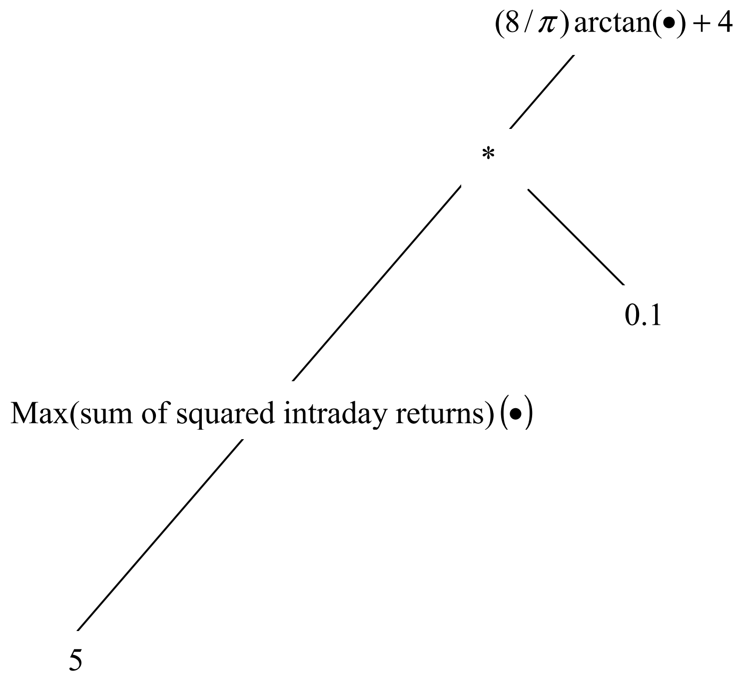


Figure 1. An example of a hypothetical forecast function

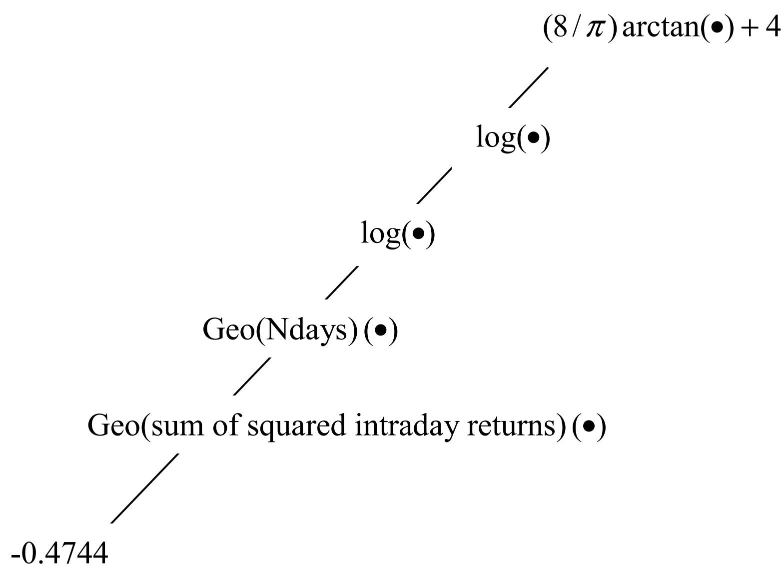


Figure 2. An example of a one-day ahead forecasting functions for the DEM found by the genetic program

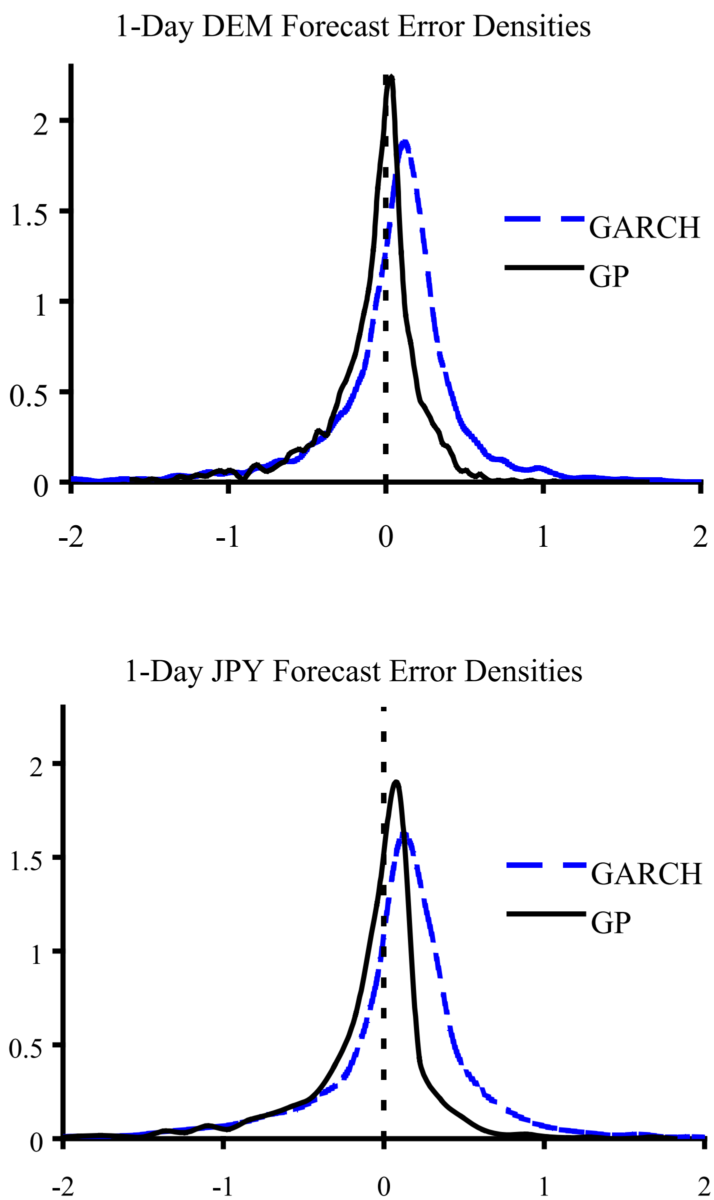


Figure 3. The kernel estimates of the densities of the one-day forecast errors (forecast minus realized volatility) for the DEM and JPY for genetic program and GARCH(1, 1) model over the out-of-sample period, December 31, 1986 to September 21, 1999. The dotted vertical line denotes zero.

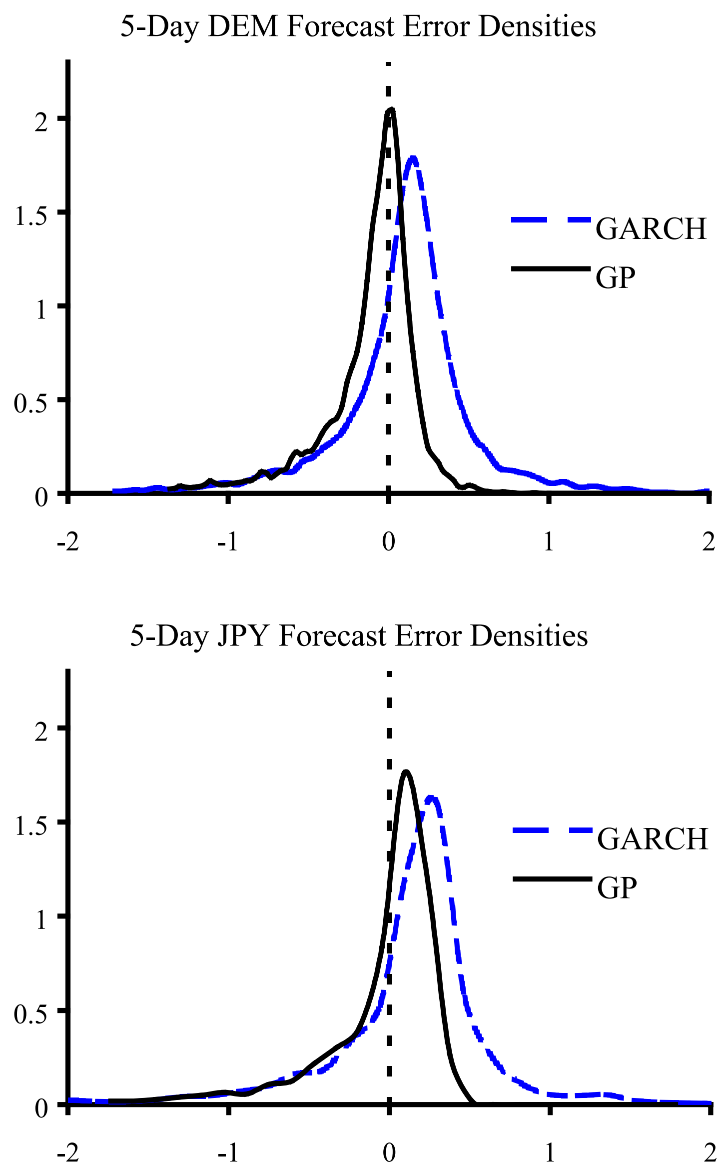
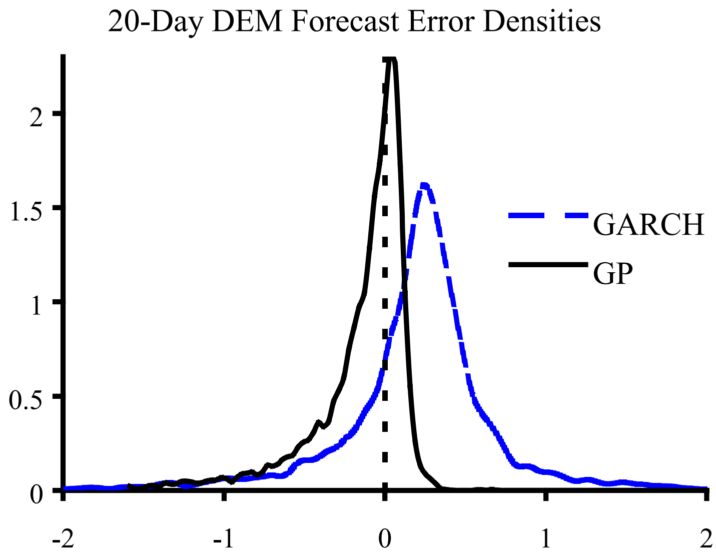


Figure 4. The kernel estimates of the densities of the five-day forecast errors (forecast minus realized volatility) for the DEM and JPY for genetic program and GARCH(1, 1) model over the out-of-sample period, December 31, 1986 to September 21, 1999. The dotted vertical line denotes zero.



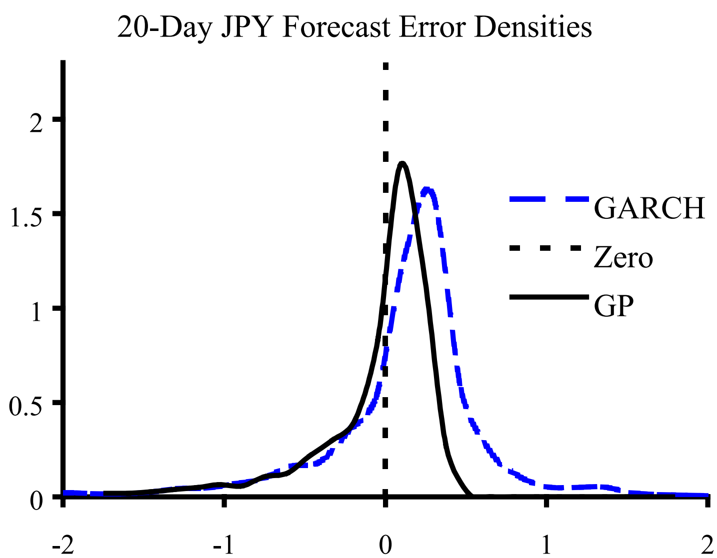


Figure 5. The kernel estimates of the densities of the twenty-day forecast errors (forecast minus realized volatility) for the DEM and JPY for genetic program and GARCH(1, 1) model over the out-of-sample period, December 31, 1986 to September 21, 1999. The dotted vertical line denotes zero.

Table 1. Data type and source

Time	Source	Type of Price
1000	Swiss National Bank	triangular arbitrage on bid rates
1400	Federal Reserve Bank of New York	mid point of bid and ask
1600	Bank of England	triangular arbitrage, unspecified
1700	Federal Reserve Bank of New York	mid point of bid and ask
2200	Federal Reserve Bank of New York	mid point of bid and ask

Table 2. In-sample comparison of genetic program, GARCH and RiskMetrics: the baseline case

Exchange Rate	horizon	MSE EW GP	MSE MW GP	MSE GARCH	MSE RM	MAE EW GP	MAE MW GP	MAE GARCH	MAE RM	R^2 EW GP	R^2 MW GP	R^2 GARCH	R^2 RM
DEM	1	0.50	0.53	0.50	0.49	0.30	0.33	0.33	0.33	0.18	0.15	0.16	0.14
DEM	5	0.56	0.59	0.56	0.52	0.31	0.34	0.37	0.34	0.12	0.11	0.10	0.10
DEM	20	0.61	0.63	0.67	0.56	0.33	0.34	0.46	0.37	0.08	0.04	0.04	0.05
JPY	1	0.56	0.58	0.60	0.62	0.32	0.32	0.38	0.37	0.22	0.20	0.14	0.08
JPY	5	0.65	0.65	0.73	0.66	0.36	0.37	0.43	0.38	0.06	0.04	0.02	0.04
JPY	20	0.66	0.67	0.71	0.69	0.38	0.39	0.51	0.40	0.05	0.03	0.01	0.02

Notes: The in-sample mean square error (MSE), mean absolute error (MAE) and R^2 from GARCH(1,1), RiskMetrics (RM) and genetic program (GP) forecasts on DEM/USD and JPY/USD data at 3 forecast horizons: 1-day, 5-days, and 20-days. The GP forecast was generated using five data functions and without a penalty for complexity. In columns 3, 7 and 11 we report the forecast statistics—MSE, MAE and R^2 —for the equally weighted (EW) genetic programming method. In columns 4, 8 and 12 we report the analogous statistics for the median weighted (MW) genetic programming forecast. Columns 5, 9 and 13 contain the results for the GARCH forecast. Columns 6, 10 and 13 contain RiskMetrics forecast statistics. The in-sample period was June 1975 to December 30, 1986.

Table 3. Out-of-sample comparison of genetic program, GARCH and RiskMetrics: the baseline case

Exchange Rate	horizon	MSE EW GP	MSE MW GP	MSE GARCH	MSE RM	MAE EW GP	MAE MW GP	MAE GARCH	MAE RM	R^2 EW GP	R^2 MW GP	R^2 GARCH	R^2 RM
DEM	1	0.35	0.38	0.33	0.32	0.30	0.34	0.33	0.32	0.09	0.08	0.12	0.10
DEM	5	0.38	0.42	0.36	0.34	0.31	0.35	0.35	0.33	0.06	0.06	0.08	0.07
DEM	20	0.41	0.42	0.44	0.37	0.31	0.31	0.43	0.35	0.01	0.01	0.02	0.02
JPY	1	1.35	1.35	1.29	1.33	0.42	0.44	0.47	0.47	0.14	0.13	0.16	0.11
JPY	5	1.48	1.48	1.56	1.44	0.43	0.45	0.52	0.49	0.03	0.02	0.04	0.06
JPY	20	1.48	1.48	1.43	1.46	0.45	0.46	0.55	0.51	0.02	0.02	0.05	0.05

Notes: The out-of-sample MSE, MAE and R^2 from GARCH(1,1), RiskMetrics (RM) and genetic program (GP) forecasts on DEM/USD and JPY/USD data at 3 forecast horizons: 1-day, 5-days, and 20-days. The GP forecast was generated using five data functions and without a penalty for complexity. The out-of-sample period was December 31, 1986 to September 21, 1999. See the notes to Table 2 for column headings.

Table 4. Out-of-sample results using the data functions *geo*, *mem,* and *arch5*

Exchange Rate	horizon	MSE EW GP	MSE MW GP	MSE GARCH	MSE RM	MAE EW GP	MAE MW GP	MAE GARCH	MAE RM	R^2 EW GP	R^2 MW GP	R^2 GARCH	R^2 RM
DEM	1	0.37	0.44	0.33	0.32	0.29	0.37	0.33	0.32	0.12	0.05	0.12	0.10
DEM	5	0.36	0.37	0.36	0.34	0.30	0.30	0.35	0.33	0.05	0.04	0.08	0.07
DEM	20	0.38	0.38	0.44	0.37	0.30	0.30	0.43	0.35	0.01	0.01	0.02	0.02
JPY	1	1.27	1.31	1.29	1.33	0.43	0.44	0.47	0.47	0.18	0.15	0.16	0.11
JPY	5	1.45	1.46	1.56	1.44	0.46	0.46	0.52	0.49	0.04	0.03	0.04	0.06
JPY	20	1.49	1.62	1.43	1.46	0.44	0.50	0.55	0.51	0.04	0.00	0.05	0.05

Notes: The out-of-sample MSE and R^2 from GARCH(1,1), RiskMetrics (RM) and genetic program (GP) forecasts on DEM/USD and JPY/USD data at 3 forecast horizons: 1-day, 5-days and 20-days. The GP forecast was generated using eight data functions including *geo*, *mem,* and *arch5* (for descriptions see equations (6) – (8) in the text) and without a penalty for complexity. The out-of-sample period was December 31, 1986 to September 21, 1999. See the notes to Table 2 for column headings.

Table 5. Tests for mean forecast bias

Exchange Rate	horizon	Mean Volatility	Predicted Volatility EW GP	Bias EW GP	Bias p-value EW GP	Predicted Volatility MW GP	Bias MW GP	Bias p-value MW GP	Predicted Volatility GARCH	Bias GARCH	Bias p-value GARCH	Predicted Volatility RM	Bias RM	Bias p-value RM
DEM	1	0.43	0.29	-0.15	0.00	0.24	-0.19	0.00	0.46	0.03	0.00	0.43	0.00	0.92
DEM	5	0.43	0.23	-0.20	0.00	0.16	-0.27	0.00	0.49	0.05	0.00	0.43	0.00	0.92
DEM	20	0.43	0.19	-0.24	0.00	0.18	-0.25	0.00	0.59	0.16	0.00	0.43	0.00	0.92
JPY	1	0.56	0.33	-0.22	0.00	0.32	-0.24	0.00	0.57	0.02	0.06	0.55	0.00	0.88
JPY	5	0.56	0.37	-0.19	0.00	0.41	-0.15	0.00	0.59	0.04	0.07	0.55	-0.01	0.87
JPY	20	0.56	0.42	-0.14	0.00	0.44	-0.12	0.01	0.65	0.09	0.02	0.55	-0.01	0.89

Notes: In column 3 mean volatility is the mean daily integrated volatility over the out-of-sample period December 31, 1986 to September 21, 1999. Columns 4, 5 and 6 report the following statistics for the equally weighted genetic programming forecasts over the same period: mean forecast of integrated volatility, the bias in the forecast (predicted volatility minus realized volatility) and the p-value for the test that the mean bias is zero. Columns 7 through 9 report the statistics for the median weighted genetic programming forecasts and columns 10 through 12 report the analogous results for GARCH forecasts. The RiskMetrics statistics are in columns 13 through 15. The genetic program forecasts are based on the 5-function model described in Table 3. The p-values are computed with Newey-West corrections for heteroskedasticity and serial correlation. The lag length was selected by the Newey and West (1994) procedure.

Table 6: Test of generalized method of second forecast error moment domination

	Eigenvalues GARCH-EW GP	Eigenvalues GARCH-MW GP	Eigenvalues RM-EW GP	Eigenvalues RM-MW GP	Eigenvalues GARCH-RM
DEM	-0.090	-0.148	-0.021	-0.037	-0.017
	0.012	-0.003	0.003	-0.138	0.086
	0.082	0.079	-0.084	-0.002	0.036
JPY	-0.369	-0.359	-0.028	-0.035	-0.295
	0.145	0.127	0.055	0.059	0.200
	0.199	0.203	-0.101	-0.102	0.144

Notes: The table provides sets of eigenvalues for the test of generalized method of second forecast error moment criterion. The first model dominates the second model if all the eigenvalues in a set are nonpositive and at least one is negative. The columns 2 through 6 provide the sets of eigenvalues for the following cases: 1) GARCH model versus the equally weighted genetic programming forecasts for the baseline case, as in Table 3; 2) GARCH model versus the median weighted genetic programming forecasts for the baseline case, as in Table 3; 3) RiskMetrics model versus the equally weighted genetic programming forecasts for the baseline case; 4) RiskMetrics model versus the median weighted genetic programming forecasts for the baseline case; 5) GARCH model versus RiskMetrics forecasts.

References

Andersen, Torben and Tim Bollerslev. 1998. "Answering the skeptics: Yes, standard volatility models do provide accurate forecasts." *International Economic Review* 39: 885-905.

Andersen, Torben, Tim Bollerslev, Francis Diebold and Paul Labys. 2001. Modeling and Forecasting Realized Volatility. NBER Working Paper 8160.

Baillie, R. and T. Bollerslev. 1989. "The Message in Daily Exchange Rates: A Conditional-Variance Tale." *Journal of Business and Economic Statistics* 7: 297-305.

Baillie, R. and T. Bollerslev. 1991. "Intra-Day and Inter-Market Volatility in Foreign Exchange Rates." *Review of Economic Studies* 58: 565-585.

Bollerslev, T. 1986. "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics* 31: 307-327.

Clements, M.P. and D.F. Hendry. 1993. "Econometric Evaluation of linear macroeconomic models." *Review of Economic Studies* 53: 671-690.

Engle, Robert F. 1982. "Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of U.K. Inflation." *Econometrica* 50: 987-1008.

<u>Genetic Programming in Computational Finance</u>, edited by Shu-Heng, forthcoming from by Kluwer publishing in 2002.

Holland, J. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.

J.P. Morgan, 1996, *RiskMetrics Technical Document, Part II: Statistics of Financial Market Returns*, 4th edition, New York.

Koza, JR. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Neely, Christopher J. 1999. “Target Zones and Conditional Volatility: The Role of Realignments.” *Journal of Empirical Finance* 6: 177-92.

Neely, Christopher J. 1999. Risk-Adjusted, Ex Ante, Optimal, Technical Trading Rules in Equity Markets, Federal Reserve Bank of St. Louis Working Paper WP 99-015C.

Neely, Christopher J. and Paul A. Weller. 1999. “Technical Trading Rules in the European Monetary System.” *Journal of International Money and Finance* 18: 429-58.

Neely, Christopher J. and Paul A. Weller. 2001. “Technical Analysis and Central Bank Intervention.” *Journal of International Money and Finance* (forthcoming).

Neely, Christopher J., Paul A. Weller, and Rob Dittmar. 1997. “Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach.” *Journal of Financial and Quantitative Analysis* 32: 405-426.

Newbold, Paul, David I. Harvey and Stephen L. Leybourne. 1999. “Ranking Competing Multi-Step Forecasts.” in Cointegration, Forecasting and Causality, Oxford University Press, New York, Robert F. Engle and Halbert White, editors.

Newey, Whitney K. and Kenneth D. West. 1987. “A Simple Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix.” *Econometrica* 55: 703-708.

Newey, Whitney K. and Kenneth D. West. 1994. “Automatic Lag Selection in Covariance Matrix Estimation.” *Review of Economic Studies* 61: 631-53.

West, Kenneth D. and Dongchul Cho. 1995. “The Predictive Ability of Several Models of Exchange Rate Volatility.” *Journal of Econometrics* 69(2): 367-91.