```
function [flag,thresh] = calcTransThresh_02(...
                exchange,...
                portfolio,...
                symbol,...
                avgWindow)
    % This function calculates
    % the buy/sell threshold
    % for a stock using a non-linear
    % strategy.  In other words,
    % the amount by which the
    % price has deviated from the
    % average WILL affect
    % the buy/sell threshold.
    % The buy/sell threshold
    % will be larger the farther
    % the price deviates from the
    % average during the last
    % "x" days.  This means the
    % simulation will buy/sell
    % more shares for each $
    % of price change as the
    % price deviates more and more
    % from the average.

    % The threshold will be given
    % in number of shares per dollar
    % to either buy or sell given
    % the change in stock price.
    % If the function succeeds,
    % flag = 1, otherwise it
    % is set to 0.

    % Get stock data from the exchange.
    [flag,stockStruct] = getStockData_exchange(...
        exchange,symbol);
    % Make sure the stock is found.
    if(flag == 0)
        % Handle error.
        fprintf('Stock NOT found in exchange!\n');
        thresh = 0;
        flag = 0;
        return;
    end
    % Get the average stock price
    % for the given stock over time,
    % which in the case of this
    % function will be given by
    % the average of the price
    % over the last "x" days.
    if(length(stockStruct.close) <= avgWindow)
        % Average all elements of
```

```
    % "close" array.
    avgPrice = mean(stockStruct.close);
else
    % Average last "x" elements
    % of "close" array.
    temp = (length(stockStruct.close) ...
            - avgWindow + 1);
    avgPrice = mean(...
        stockStruct.close(temp:end));
end;
% Get the number of shares
% currently owned for the
% desired symbol.
temp = strcmp(portfolio.stockSymbols,symbol);
I = find((temp == 1),...
        1,'first');
currentShares = portfolio.stockShares(I);
% Determine the buy/sell threshold
% by dividing the current number
% of shares purchased by the average
% price over the last "x" days.
% This threshold is chosen
% based on the assumption that the
% buy and sell thresholds will be
% equal.  Thus, the threshold
% calculated here represents the
% amount of price change that would
% have to occur to either wipe out
% or double the number of shares
% owned relative to the current number
% of shares owned, assuming
% the price changes for those two
% scenarios would be equal.
thresh = (currentShares / avgPrice);

% Note that this threshold represents
% a continuous slope (non-linear) of
% shares vs. price.
% In other words, going strictly by this
% number, an infinitesimal price change
% will warrant the purchase/sale of
% a corresponding infinitesimal number
% of shares.  This is unrealistic
% because, firstly, transactions take
% time to execute, and if a
% transaction were to be required
% for every infinitesimal price change,
% then the transactions would be
% required to execute
% instantaneously.  Also, because
% of trade commissions, this buy/sell
```

```
    % strategy would be not cost effective
    % by a long shot.  In order to deal
    % with this issue, the continuous-slope
    % threshold will be used as a baseline
    % threshold, and then an additional
    % calculation will take into account
    % a fixed trade commission.  A
    % transaction will not be made unless
    % enough profit would be made on the
    % sale to cover the commission.  If
    % the transaction is a buy, the same
    % threshold will be used as if it were
    % a sell, since money is always lost
    % on a buy transaction.  Essentially,
    % the model will traverse the
    % shares/price slope calculated here
    % until it makes sense to actually
    % execute a transaction.

    % Return success flag.
    flag = 1;

    return;
end
```