

```
function [flag,thresh] = calcTransThresh_01( ...
    exchange, ...
    portfolio, ...
    symbol)
% This function calculates
% the buy/sell threshold
% for a stock using a linear
% strategy. In other words,
% the amount by which the
% price has deviated from the
% average will NOT affect
% the buy/sell threshold.
% No matter how high or low
% the price goes, the same
% number of shares will be
% bought/sold per $ of change.

% This function assumes that
% the long term average of the
% stock in question is equal to
% the price of the stock when
% the first shares were first
% purchased (on day zero).

% 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 price when the stock was
% initially purchased on day zero.
avgPrice = stockStruct.close(1);
% Get the initial number of
```

```
% shares from the initial purchase
% on day zero.
for i = (1:size(portfolio.transactions,1))
    if((portfolio.transactions{i,4} == 0) ...
        || (portfolio.transactions{i,4} == 1)) ...
        && strcmp(portfolio.transactions{i,8},symbol))
        startShares = portfolio.transactions{i,10};
        break;
    end
    % If control reaches this point,
    % the first-day transaction for
    % the desired symbol was not found.
    % Handle error.
    fprintf('Start shares not found in portfolio!\n');
    flag = 0;
    thresh = 0;
    return;
end
% Determine the buy/sell threshold
% by dividing the original number
% of shares purchased by the initial
% price when the shares were first
% purchased. 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 original number
% of shares owned, assuming
% the price changes for those two
% scenarios would be equal.
thresh = (startShares / avgPrice);

% Note that this threshold represents
% a continuous slope 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
```