```
function [portfolioMod,accountMod] = updatePortfolio( ...
                        exchange, ...
                        portfolio, ...
                        account, ...
                        currentTime, ...
                        tradeCommission, ...
                        minTransProfit, ...
                        avgWindow)
    % This function will update the
    % given portfolio based on
    % simulated market activity.

    % In the end, after all
    % calculations have been done,
    % there will be an action determined
    % for each stock in the exchange.
    % The action will be in the following
    % form:
    %   1. Stock symbol
    %   2. Number of shares to act on
    %   3. Operation (buy/sell)
    % If the number of shares is zero,
    % this means "do nothing."

    % After it is determined what actions
    % will be taken on each stock in the
    % exchange, the changes will be made
    % using the "buy" and "sell" functions,
    % and the transactions will all be
    % recorded to the transaction list
    % by those same functions.

    % Loop through all stocks in the
    % exchange and decide how to
    % transact for each stock.
    for i = (1:length(exchange))
        % If the stock price is zero,
        % don't do anything.  Assume the
        % company has gone out of
        % business.  Skip ahead to
        % the next iteration of the loop.
        if(exchange(i).currentPrice == 0)
            continue;
        end
        % Retrieve the information
        % about the current stock from
        % the portfolio.
        [flag,currentStockPortfolioData] = ...
            getStockData_portfolio( ...
            portfolio,exchange(i).symbol);
        % Check the flag to make
```

```
% sure the function executed
% successfully.
if(flag == 0)
    % Stock not found in portfolio.
    % Handle error.
    fprintf('Stock not found in portfolio!\n');
    portfolioMod = portfolio;
    accountMod = account;
    return;
end

% The stock was found in the
% portfolio.  Continue with
% normal buy/sell decision
% making.

% Calculate the buy/sell
% threshold for the current stock.
[flag,thresh] = calcTransThresh_01( ...
    exchange,portfolio, ...
    exchange(i).symbol);     %,...
    %avgWindow);
% Check the flag to make
% sure the function executed
% successfully.
if(flag == 0)
    % Handle error.
    fprintf('Error calculating transaction threshold!\n');
    portfolioMod = portfolio;
    accountMod = account;
    return;
end
% Determine the stock price
% when the last transaction
% was executed.
lastTransPrice =...
    currentStockPortfolioData.transactions{end,9};
% Use the threshold ratio to
% determine how many shares of
% each stock to buy or sell.
% Account for the fixed
% trade commission from the
% broker.  The decision to
% buy or sell is based on the
% price change since the last
% transaction for this stock.

% Calculate price change.
priceChange = (exchange(i).currentPrice ...
    - lastTransPrice);
% Decide whether to buy or sell.
```

```
if(priceChange > 0)
    % Sell.
    transType = 'SELL';
elseif(priceChange < 0)
    % Buy.
    transType = 'BUY';
else
    % Do nothing.
    transType = 'NONE';
end
% Calculate how many shares
% to transact based only on the
% price change and threshold
% slope.  Round this number
% down to the nearest integer
% so the number of shares
% transacted is a whole number.
sharesToTrans = (abs(priceChange) * thresh);
sharesToTrans = floor(sharesToTrans);

% See if the proposed transaction
% will produce a portfolio
% value increase to cover the
% commission plus the desired
% profit per sale, set in the
% "parameters" script.  Pretend
% that either a buy or a sell is
% a sell.  This way, we can
% intuitively decide whether to
% transact based on profit, and
% then just mirror that decision
% and apply it to a buy as well.

% Calculate the gain/loss
% magnitude for the proposed
% transaction.
sharesOwn = currentStockPortfolioData.numShares;
gainLossMag = abs((sharesOwn ...
    * exchange(i).currentPrice) ...
    - (sharesOwn * lastTransPrice));
% If the magnitude of the
% gain warrants a sale,
% then execute the transaction.
% Since we are using absolute
% value, the decisions to buy
% or sell are both based on the
% magnitude of the gain, even
% if it is negative.  Since it
% doesn't make sense to calculate
% the buy threshold based on how
% much will be lost from the buy,
```

```
        % we simply mirror the sell
        % threshold and use it for buying
        % as well.
        if(gainLossMag >= (tradeCommission ...
                        + minTransProfit))
            % Yes, go ahead with
            % transaction.
            if(strcmp(transType,'SELL'))
                % Attempt to execute sale.
                [flag,msg,portfolio,account] = ...
                    sell(portfolio, ...
                    exchange, ...
                    account, ...
                    exchange(i).symbol, ...
                    sharesToTrans, ...
                    tradeCommission, ...
                    currentTime);
                if(flag == 1)
                    % Success!
                elseif(flag == 0)
                    % Try selling again
                    % with all shares owned.
                    % The sell function
                    % will terminate if
                    % it attempts to sell
                    % more shares than are
                    % owned.
                    [flag,msg,portfolio,account] = ...
                        sell(portfolio, ...
                        exchange, ...
                        account, ...
                        exchange(i).symbol, ...
                        sharesOwn, ...
                        tradeCommission, ...
                        currentTime);
                    if(flag == 1)
                        % Success!
                    elseif(flag == 0)
                        % Can't sell.
                        % Do nothing.
                    else
                        % Handle error.
                    end
                else
                    % Handle error.
                end
            elseif(strcmp(transType,'BUY'))
                % Attempt to execute buy.
                [flag,msg,portfolio,account] = ...
                    buy(portfolio, ...
                    exchange, ...
```

```
                    account, ...
                    exchange(i).symbol, ...
                    sharesToTrans, ...
                    tradeCommission, ...
                    currentTime);
                if(flag == 1)
                    % Success!
                elseif(flag == 0)
                    % Failed to buy.
                    % This should never
                    % happen unless the
                    % stock is not in the
                    % exchange, however that
                    % wouldn't be possible
                    % if control got to this
                    % point.
                    % Handle error.
                    fprintf('Failed to buy stock!\n');
                    portfolioMod = portfolio;
                    accountMod = account;
                    return;
                else
                    % Handle error.
                end
            elseif(strcmp(transType,'NONE'))
                % Do Nothing.
            else
                % Handle error.
            end
        else
            % No, don't go ahead
            % with transaction.
        end
    end

    % Make sure to return the
    % modified portfolio so the
    % data is not lost.
    portfolioMod = portfolio;
    accountMod = account;

    return;

end
```