## Training Data Generation

**Current Idea**

1. Given a set of parameters for GARCH (in Physical measure)
2. Given the initial asset price $S_0$, use Monte Carlo method to simulate a path of asset prices, $S_1, S_2, ...S_N$, with say $N = 500$ (Under **P** measure)
3. Select last 30-50 days on the path, for each day, use the selected asset price (under **Q**) as the initial price to generate American option prices with various strike prices (11-17) and maturities (7 days to 1 year). **Pay attention to the transformation from the physical measure to the risk-neutral measure**.

**Pseudo Code**

```
% Pseudo Code of Pricing American Options By the Willow Tree and Monte Carlo Method
% 1. Initialize Option Parameters
T = ...; % Time to maturity
K = ...; % Strike price
r = ...; % Risk-free rate
S0 = ...; % Initial asset price
N = ...; % Number of time steps for simulation (ex. 500)
M = ...; % Number of Monte Carlo paths
delta = T/N; % Time step
h0 = ...; % Initial volatility

% 2. Initialize HN-GARCH parameters under P Measure
alpha = ...;
beta = ...;
omega = ...;
gamma = ...;
lambda = ...;

% 3. Simulate paths using Monte Carlo simulation under P measure
% Number of paths will be N so we simulate a path of asset prices
% from S1, S2, ... S_N
points = N + 1; % number of points

Z = randn(points + 1, N);
Z1 = randn(points, N);
ht = nan(points + 1, N);
ht(1,:) = h0*ones(1,N);
Xt(1,:) = log(S0)*ones(1,N);
for i=2:points
    ht(i,:) = omega+alpha*(Z(i-1,:)-gamma*sqrt(ht(i-1,:))).^2+beta*ht(i-1,:);
    Xt(i,:) = Xt(i-1,:)+(r-0.5*ht(i,:))+sqrt(ht(i,:)).*Z(i,:);
end
ht(i+1,:) = omega+alpha*(Z(i,:)-gamma*sqrt(ht(i,:))).^2+beta*ht(i,:);
S = exp(Xt);

% 4. Risk-neutralize GARCH parameters (Q measure)
eta = ...;
omega_Q = omega / (1-2*alpha*eta);
gamma_Q = gamma*(1-2*alpha*eta);
alpha_Q = alpha /  (1-2*alpha*eta)^2;
lambda_Q = lambda*(1-2*alpha*eta);
rho_Q = lambda_Q + gamma_Q + 1/2;
```

```
% 5. Initialize Willow Tree parameters
m_h = ...;
m_ht = ...;
m_x = ...;
gamma_h = ...;
gamma_x = ...;

% 6. Construct the willow tree for ht (using Q measure parameters)
[hd, qhd] = genhDelta(h0, beta_Q, alpha_Q, gamma_Q, omega_Q, m_h, gamma_h);
nodes_ht = TreeNodes_ht_HN(m_ht, hd, qhd, gamma_h, alpha_Q, beta_Q, gamma_Q, omega_Q,
N + 1);
[P_ht_N, P_ht] = Prob_ht(nodes_ht, h0, alpha_Q, beta_Q, gamma_Q, omega_Q);

% 7. Construct the willow tree for Xt (using Q measure parameters)
[nodes_Xt, mu, var, k3, k4] = TreeNodes_logSt_HN(m_x, gamma_x, r, hd, qhd, S_0,
alpha_Q, beta_Q, gamma_Q, omega_Q, N);
[q_Xt, P_Xt, tmpHt] = Prob_Xt(nodes_ht, qhd, nodes_Xt, S_0, r, alpha_Q, beta_Q,
gamma_Q, omega_Q);
nodes_S = exp(nodes_Xt);

% 8. Generate Data for last number of days
days_to_price = ...; % could be 50 for the last 50 days
moneyness = [0.8, 0.9, 0.95, 1, 1.05, 1.1, 1.2]; % Example: from 80% to 120% of
current price
maturities = [7/365, 1/12, 1/4, 1/2, 1]; % 7 days, 1 month, 3 months, 6 months, 1
year

A_prices = zeros(days_to_price, length(moneyness), length(maturities));
A_sig = zeros(days_to_price, length(moneyness), length(maturities));
A0_prices = zeros(days_to_price, length(moneyness), length(maturities));

for i = 1:days_to_price
    % Use the last simulated path as an example
    S_t = simulated_paths{1}(end-days_to_price+i);
    % Calculate strike prices based on current stock price
    strike_prices = S_t * moneyness;
    for j = 1:length(moneyness)
        for k = 1:length(maturities)
            CorP = -1; % Call or Put
            [A_sig(i,j,k), A_prices(i,j,k), A0_prices(i,j,k)] = impVol_HN(r,
lambda_Q,
                                ... omega_Q, beta_Q, alpha_Q, gamma_Q, h0, S_t,
                                ... strike_prices(j), maturities(k), N, m_h, m_x,
CorP);
        end
    end
end
```

Functions to be aware of:

- `genhDelta`: Generates the discrete values and probabilities of a std normal distribution that are used to construct a Willow tree for the conditional variance in the HN model.
- `TreeNodes_ht_HN`: Constructs the Willow tree for the conditional variance in the HN model.
- `Prob_ht`: Calculates the transition probabilities of the nodes in the Willow tree for the conditional variance in the HN model.
- `TreeNodes_logSt_HN`: Constructs the nodes of the Willow tree for the log asset price in the HN model, as well as the first four moments.
- `Prob_Xt`: Calculates the transition probabilities of the nodes in the Willow tree for the log asset price in the HN model.
- `impVol_HN`: Calculates the American option price, the implied volatility, and the option price using the model parameters in the HN model.

**f_hhh Situation** The current dependency on f_hhh relies on a MEX file which is compiled from C code, currently the best tool to handle this is `mkoctfile` from Octave. Currently a Matlab native code is giving the following results (from the original demo code Prices)

| Windows f_hhh.mexw64 | Linux f_hhh.m | Linux f_hhh.mexa64 | Rel Error of f_hhh.m to Mex |
|---|---|---|---|
| 2.72207972210372 | 25.9570397649806 | 2.68249096732144 | 853.57% |
| 10.0008627425866 | 35.6958936816417 | 10 | 256.93% |
| 20 | 45.4628396735826 | 20 | 127.31% |
| 30 | 55.2436491040201 | 30 | 84.15% |
| 40 | 65.0342712634885 | 40 | 62.59% |
| 50 | 74.848603483338 | 50 | 49.70% |
| 60 | 84.6710358855933 | 60 | 41.12% |
| 70 | 94.4938124800953 | 70 | 34.99% |
| 80 | 104.317906789725 | 80 | 30.40% |
| 90 | 114.142001099355 | 90 | 26.82% |
| 100 | 123.966095408986 | 100 | 23.97% |
| 110 | 133.790189718616 | 110 | 21.63% |
| 120 | 143.614284028246 | 120 | 19.68% |
| 130 | 153.438378337876 | 130 | 18.03% |
| 140 | 163.262472647506 | 140 | 16.62% |
| 150 | 173.086566957136 | 150 | 15.39% |
| 160 | 182.910661266766 | 160 | 14.32% |
| 170 | 192.735598373633 | 170 | 13.37% |
| 180 | 202.561343435829 | 180 | 12.53% |
| 190 | 212.387088498024 | 190 | 11.78% |
| 200 | 222.212833560219 | 200 | 11.11% |
| 210 | 232.038578622415 | 210 | 10.49% |
| 220 | 241.86432368461 | 220 | 9.94% |
| 230 | 251.690068746806 | 230 | 9.43% |
| 240 | 261.516107218318 | 240 | 8.97% |
| 250 | 271.34349038818 | 250 | 8.54% |
| 260 | 281.170873558042 | 260 | 8.14% |
| 270 | 290.998256727904 | 270 | 7.78% |
| 280 | 300.825639897766 | 280 | 7.44% |
| 290 | 310.653023067628 | 290 | 7.12% |
| 300 | 320.481302750366 | 300 | 6.83% |