

Daten aus der Excel-Sheet "Dateset.xlsx":

CounterpartyID	NettingID	Principal	Maturity	LegType	LegRateReceiving	LegRatePaying	LatestFloatingRate	Period
1		100000	736678	1	0,0423	20	0,033	1

Der Code für diese Datei:

```
swapFile = 'Dateset.xlsx';
swapData = readtable(swapFile, 'Sheet', 'Tabelle1');

swaps = struct( ...
    'Counterparty', [], ...
    'NettingID', [], ...
    'Principal', [], ...
    'Maturity', [], ...
    'LegRate', [], ...
    'LegType', [], ...
    'LatestFloatingRate', [], ...
    'FloatingResetDates', []);

swaps.Counterparty = swapData.CounterpartyID;
swaps.NettingID = swapData.NettingID;
swaps.Principal = swapData.Principal;
swaps.Maturity = swapData.Maturity;
swaps.LegType = [swapData.LegType ~swapData.LegType];
swaps.LegRate = [swapData.LegRateReceiving swapData.LegRatePaying];
swaps.LatestFloatingRate = swapData.LatestFloatingRate;
swaps.Period = swapData.Period;
swaps.LegReset = ones(size(swaps.LegType));

numSwaps = numel(swaps.Counterparty);

Settle = datenum('14-Dec-2007');

Tenor = [3 6 12 5*12 7*12 10*12 20*12 30*12]';
ZeroRates = [0.033 0.034 0.035 0.040 0.042 0.044 0.048 0.0475]';

ZeroDates = datemnth(Settle, Tenor);
Compounding = 2;
Basis = 0;
RateSpec = intenvset('StartDates', Settle, 'EndDates', ZeroDates, ...
    'Rates', ZeroRates, 'Compounding', Compounding, 'Basis', Basis);
```

```

figure;
plot(ZeroDates, ZeroRates, 'o-');
xlabel('Date');
datetick('keeplimits');
ylabel('Zero rate');
grid on;
title('Yield Curve at Settle Date');

% Number of Monte Carlo simulations
numScenarios = 1000;

% Compute monthly simulation dates, then quarterly dates later.
simulationDates = datemnth(Settle,0:12);
simulationDates = [simulationDates datemnth(simulationDates(end),3:3:74)'];
numDates = numel(simulationDates);

floatDates = cfdates(Settle-360,swaps.Maturity,swaps.Period);
swaps.FloatingResetDates = zeros(numSwaps,numDates);
for i = numDates:-1:1
    thisDate = simulationDates(i);
    floatDates(floatDates > thisDate) = 0;
    swaps.FloatingResetDates(:,i) = max(floatDates,[],2);
end

Alpha = 0.2;
Sigma = 0.015;

hw1 = HullWhite1F(RateSpec,Alpha,Sigma);
% Use reproducible random number generator (vary the seed to produce
% different random scenarios).
prevRNG = rng(0, 'twister');

dt = diff(yearfrac(Settle,simulationDates,1));
nPeriods = numel(dt);
scenarios = hw1.simTermStructs(nPeriods, ...
    'nTrials',numScenarios, ...
    'deltaTime',dt);

% Restore random number generator state
rng(prevRNG);

% Compute the discount factors through each realized interest rate
% scenario.
dfactors = ones(numDates,numScenarios);
for i = 2:numDates
    tenorDates = datemnth(simulationDates(i-1),Tenor);
    rateAtNextSimDate = interp1(tenorDates,squeeze(scenarios(i-1,:,:)), ...
        simulationDates(i),'linear','extrap');
    % Compute D(t1,t2)
    dfactors(i,:) = zero2disc(rateAtNextSimDate, ...
        repmat(simulationDates(i),1,numScenarios),simulationDates(i-1),-1,3);
end
dfactors = cumprod(dfactors,1);

i = 20;

```

```

figure;
surf(Tenor, simulationDates, scenarios(:, :, i))
axis tight
datetick('y', 'mmmyy');
xlabel('Tenor (Months)');
ylabel('Observation Date');
zlabel('Rates');
ax = gca;
ax.View = [-49 32];
title(sprintf('Scenario %d Yield Curve Evolution\n', i));

values = hcomputeMTMValues(swaps, simulationDates, scenarios, Tenor);

i = 34;
figure;
plot(simulationDates, values(:, :, i));
datetick;
ylabel('Mark-To-Market Price');
title(sprintf('Swap prices along scenario %d', i));

figure;
totalPortValues = squeeze(sum(values, 2));
plot(simulationDates, totalPortValues);
title('Total MTM Portfolio Value for All Scenarios');
datetick('x', 'mmmyy')
ylabel('Portfolio Value ($)')
xlabel('Simulation Dates')

```

Ergebnis

