Simulating the Dynamics of Housing Bubbles

Peter O'Connor

December 17, 2011

Abstract

In this project we have developed an agent based model with the objective of simulating the bubble building and bursting behaviour in the housing market. We create an agent based model, consisting of HomeOwners, Banks, Houses, and Mortgages. With this model we observed the formation of price bubbles.

0.1 Introduction

The American sub-prime mortgage crisis was a prime example of how a variety of agents, each acting for their own benefit, can create an environment in which everyone loses. It is a case of bubble economics, in which the value assigned to an asset is wildly inflated (beyond it's ill-defined "intrinsic value") based on the assumption that this asset will be worth even more in the future.

It may well be impossible to spot bubbles in advance, and to identify their size. Even if one believes they are in a bubble, it may be irrational not to invest, and therefore contribute to the bubble, so long as the investor believes they can escape before the bubble pops. Understanding the mortgage bubble, and those like it, may help shape economic policy such that these bubbles and their eventual "pops" have a less devastating effect on the economy.

The original goal of this project was to construct a model of the system that caused the American mortgage crisis. As expected, this proved to be more difficult than expected. A key component of the mortgage crisis was the repackaging of mortgages into asset-backed securities, which were split into three "tranches" based on risk levels, and sold off to investors. To the investors, these investments were quite abstracted fom the original source of income, and so they relied on credit-rating agencies to quantify the risk. House prices had been continuously rising for a long time then, and thus were regarded as a safe investment. Meanwhile, banks and other mortgage lenders were becoming increasingly risky with their lending policies, in large part because the collateral -a house- seemed almost guaranteed to increase in value, so that even in the event of forclosure, the lender could make a profit. These aggressive lending policies eventually lead to "predatory lending" practaces, inwhich loans were made with no expectation that the borrower would be able to repay, with the intention of collecting on the collateral, as well as the interest payments made in between. This famously culminated in "ninja" (No Income No Job no Assets) loans, so called because, in addition to the acronym, borrowers would sneak out in the night when they could no longer afford the mortgages. These types of loans were able to exist because, when house prices were increasing, owners could always "cash-out" on the increased equity by refinancing - that is - getting a new mortgage for the new house value, using it to pay off the old one, and pocketing the difference. This type of loan only works if house prices keep increasing. Eventually the number of forclosures resulting from this type of behaviour caused an excess of houses on the market, and therefore a drop in prices. Banks found themselves stuck with forclosed-on houses that were declining in value. Investors quickly caught on, and started dumping mortgage-backed securities. Banks found themselves in the position of receiving far less income from mortgage repayments, and being unable re-sell them. The initial downturn triggered a positive-feedback effect, causing house prices to fall signficantly. Since banks lost much of their main income source, credit became scarse.

Now since modelling this whole chain of events would be out of the scope of this project, we focus on just the housing side of things. What we're looking for is this positive-feedback effect. The initial rise, caused by the fact that part of a HomeOwner's valuation of a house is its future resale value, which in turn is predicted by its past rate of increase, followed by a max-out, and a decline.

0.2 Model

The model consists of 6 type of objects (in addition to the one simulation class). The main actor is the HomeOwner, who makes decisions about whether about what to do in each time-step. The model is initiated with a number of HomeOwners, each equipped with a set of rules to help them profit in this world. HomeOwners can either own a house or rent (or, worst comes to worst, end up homeless). While the term “HomeOwner” is used throughout the code, and will be used here for the sake of consitency, “potental HomeOwner” might be a more apt way to describe them. A diagram of the model is shown in Figure 1.

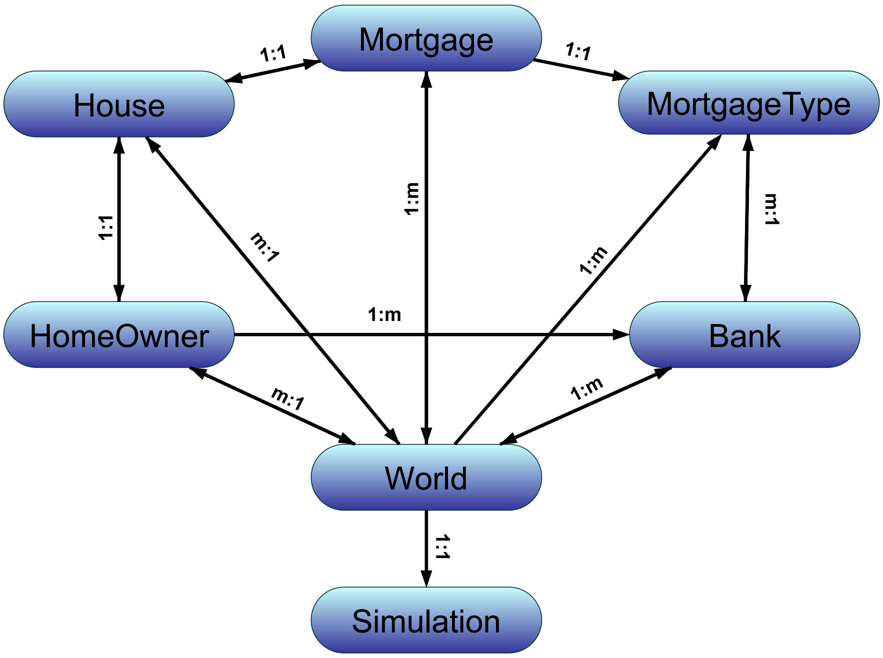

Figure 1: Top-Level view of the Model classes and how they’re interlinked. The 1’s and m’s indicate the linking between classes (eg 1:m means one-to-many, etc)

0.2.1 World

The properties and methods of the world basically encompass all those things that should be external to any of the agents. This includes: federal bank interest rate, job loss and gain rates, cost of rent (for simplicity, rent is the same for everybody), the rate at which people move/die, represented by the probability of any given HomeOwner selling the property in a given step, the house Index, and house index history, which record the ratio of house prices at time of sale to the house worth, and a list of each month’s sales, which is compiled into the house Index at the end of the month.

0.2.2 HomeOwner

The HomeOwner is the main actor in this model. He brings in a certain salary, defined in the initialization, when he has a job. The transition probabilities between haveing and not haveing a job are defined by the pJobGain and pJobLoss properties of World class. In the current version, all actions are initiated by this class. Each turn, the HomeOwners (selected in random order) look into the future an pick from a sequence of actions. There are 7 possible actions that can be taken:

1. **Keep**: Keep the house he has already, paying the mortgage bills if necessary.

2. **Default**: Walk away from the house. A person defaults when they cannot make their mortgage payments. When this happens, the house is transferred to the Bank to which the mortgage was attached, and the mortgage is removed.

3. **Sell**: Put the current house up for sale. The initial sale price is the house value times some markup (HomeOwner.saleMarkup) and it decays exponentially as the house waits longer and longer to be sold (HomeOwner.salePanic). The house price reaches a minimum at a a value determined by the HomeOwner's available cash plus the principal left on the mortgage - as selling the house for any less than that would result in the HomeOwner having negative money, which is not allowed.

4. **Refinance**: Try to refinance the current house. This means getting a new mortgage with a higher principal on the house. The HomeOwner can then use it to pay off the old mortgage, and pocket the difference. This is useful when one needs cash, and the equity of their house has grown since the purchase date. It is equivalent to buying the house off of yourself with a new mortgage.

5. **Buy**: Buy a new house. A house is decided upon (how is described in the following paragraphs), and the HomeOwner gets a mortgage to finance it. The original owner is paid by the bank providing the mortgage, so that he in turn is able to pay off his old mortgage.

   The values of the house and the mortgage are bost decided in this method (in the case of the mortgage, the principal at the end of the N steps is taken as a lump sum, and future payments ignored - the assumption being that you could if you wanted sell the house at that point.

   Finally, the equation deciding whether to buy a house is

   $$V = predictedHouseValue(Nsteps) - predictedMortageCost(Nsteps) + rentCost(Nsteps) + americanDreamValue * HouseWorth$$

   The last value in there is a quantifier for the desire to own a house. Its is there to represent the fact that people generally do not look at houses purely as investment decisions, but that there is some inherent value to owning a house.

If the above equation comes out positive, the decision is made to buy the house.

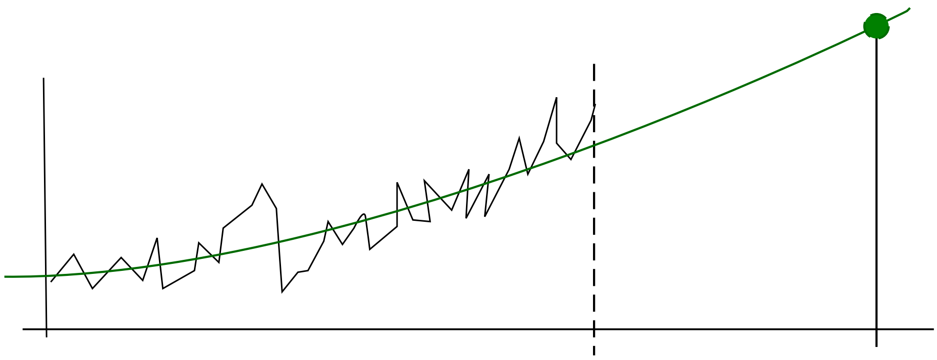

Figure 2: Predicting the value of an asset. The green line shows the exponential slope extracted from the curve.

6. **Rent**: People who have no homes will either rent or be homeless. Due to the unpleasantness of being homeless, people who can affort rent, as a rule, will rent.

7. **Be Homeless**: People who do not have a house and cannot affort to pay rent are homeless.

The decision process for a House-Owning HomeOwner is shown in Figure 4, and goes something like this: If he has already paid off the mortgage, he may still want to sell. If his mortgage is already paid off, he can choose to stay or sell. The probability of these events depends on the pSell parameter of the World object. Otherwise, his first concern is whether he can make the present payment, and whether the Loan-To-Value ratio (LTV) has exceeded some threshold (we generally use 1.2). The LTV is the ratio of the outstanding principal on the mortgage to the present value of the house. When LTV is greater then 1, it means that if you want to keep the house, you're going to have to pay mort than its presently worth, plus the interest. LTV's greater than 1 generally happen after a dip in house prices. So if these conditions are unsatisfactory, he defaults. Othwise we move on. If the house was already for sale last turn, we decrement the price and continue to sell it. Otherwise , the HomeOwner checks the short term horizon (typically 1 year-12 steps) to see if he's expecting to run out of money If so, he will first see if refinancing can help him. If not, he'll sell the house.

If the HomeOwner has no home, he looks for one. The buying process consists of selecting several houses at random, and applying several banks for finance. This is quite likely a good deal more mortgage applications that one would make in real life, but our agents don't have the benefit of a gut feeling about which ones will work out and which won't so this compensates to a certain degree. The banks will approve if the HomeOwner meets the criteria for a given mortgage. This means being able to pay the downpayment and having a salary that is a certain proportion of the initial mortgage payments. Once they've been approved the HomeOwner get to make a decision about which house/mortgage combination to pick. This is somewhat of a complex decision, as the agent must make some prediction as to what values an asset have in the future in order to determine the present value. The decision goes like this: Suppose you are trying to price a house. You have its value today, and you're trying to predict its value in N time steps. Figure 2 shows an image of this situation. We determine the geometric mean of the growth rate of the time series, rated by recentness (so more recent points count for more), then forward project N time steps. This is then turned into a present value by considering the discount rate. An important consequence of this approach is that assets that have been growing in value for a long time will have a higher predicted value.

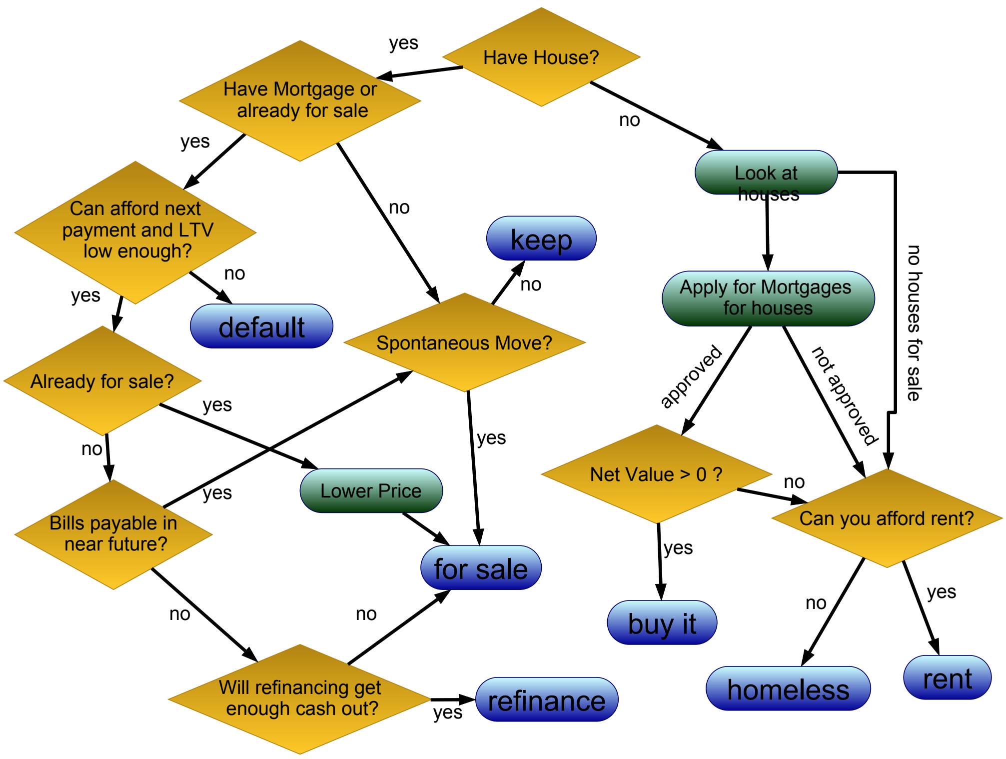

Figure 3: Logic for the HomeOwner agent.

0.2.3 House

Houses can be owned by HomeOwners or banks (they are transferred to banks after defaults). Every house has a "worth", a relative indicator of it's quality that hovers around 1. The house's value is determined by the worth time the current housing index. The house's price, which is set by the Owner at time of sale, is generally quite close to the value, plus or minus a markup or discount. Houses can be connected by mortgages to banks.

0.2.4 Mortgage and MortgageType

The mortgage is a payment plan linking a HomeOwner and a Bank. The mortgage has an associated MortgageType. Here we have two: fixed-rate and adjustable. As in real life, fixed rate mortgages require a larger down-payment, and have a higher, but guaranteed, interest rate. Adjustable-rate mortgages generally have an inital teaser period, with low interest rates, plus an interest rate that is tied to the Federal plus some premium. The idea is that a adjustable rate mortgage is likely to be cheaper, but contains more rist because of its dependence on interest rate changes.

Some important properties of our mortages are:

1. **principal**: The remaining captical cost to pay off.
2. **scheduleP**: The principal payment schedule. Each payment is a mixture of principal and interest. Knowing the principal payment schedule allows you to determin how much equity you'll have in a house after a given amount of time. All payments in the schedule will sum up to the purchase price of the house.
3. **scheduleC**: The total montly payments. For adjustable-rate mortgages, these are just an estimate, as interest is calculated on the fly accorting to the Federal bank rate.
4. **rate**: Interest rate. For fixed, it's the full interest rate, for adjustable, it's the premium over the fed.
5. **teaser**: A feature of adjustable rate mortgages, the teaser is a vector of low interest rates that you are charged at the beginning of a mortgage. This is often done as a promotion and was important in the Mortgage crisis because it attracted short-sighted, excessively optomistic, and occasionally ignorant borrowers who would not be able to make payments once the interest adjusted back to it's regular value.
6. **downPayment**: The advance-payment required to secure certain types of mortgages.

0.2.5 Bank

The simulations have the potential to include several competing banks, but for the purposes of this project they were all the same. Banks have a list of MortgageTypes from which they can create Mortgages.

0.3 Initializing

Since there were such large number of parameters for this simulation, it would have been impossible to tinker with them all until our model started working as desired. We fealt it would be unfruitful to try and match these to real-world values, the model is already such an abstraction of the real situation, matching the data precisely would have been silly. The only way to go about things was to pull them from "common-sense" values. Interest rates, salaries, house prices, etc, were all selected more or less on gut instinct.

Certain values, for instance cash and salary made sense to be correlated. These were pulled from joint gaussian distributions, with skew added later.

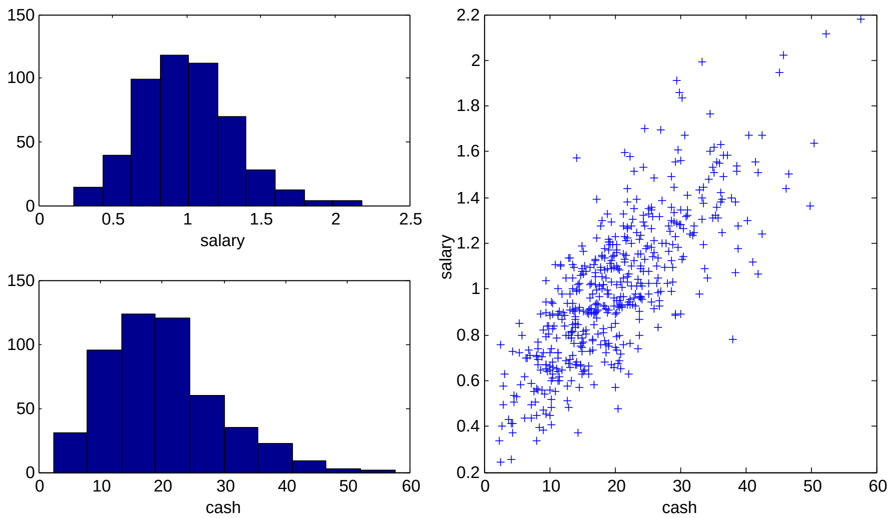

Figure 4: Initial aash and salary pulled from a joint normal distribution then squared.

0.4 Experiment

Our experiment used 500 HomeOwners, 400 Houses and 6 Banks. In the experiment we observed some interesting dynamics arise. And the system started to behave like real economic data, with "bubbles" forming and being destroyed. It appeared that these fluctuations were driven by the feedback effect that we've been discussing. Look for instance, at Figure 6. We see an initial increase, followed by a drop in the house index. The drop, it appears, makes people want to sell their homes. At about month 20 people suddenly start selling their homes. This, as we can see from 4, happens when people discover they're going to run out of money. A few defaults causes the fraction of HomeOwners with houses to dip slighltly, but this is recovered as other purchase the houses back from the bank.

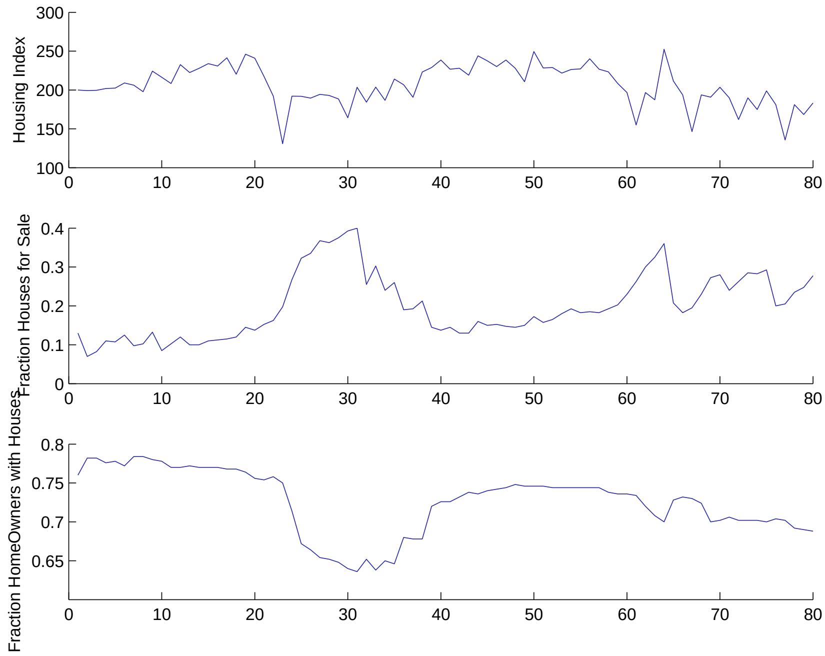

Figure 5: Top: The Housing Index: Defined as House Selling price divided by house worth. Middle: The fraction of all houses that were listed for sale. Bottom: The Fraction of HomeOwners who own houses

We can compare this with Figure ??. Here we see that the number of people holding on to their houses did indeed decrease around period 20. These people became sellers. The buyers (mustard-coloured) didn't arrive until period 30-40, presumably the time when the sellers lowered their prices to an acceptable level. Following this, we see the the market bounces back from steps 35-45.

What seems to be happenning is that the higher income earners have bought the houses out from those who had less (whose money ran out around step 20).

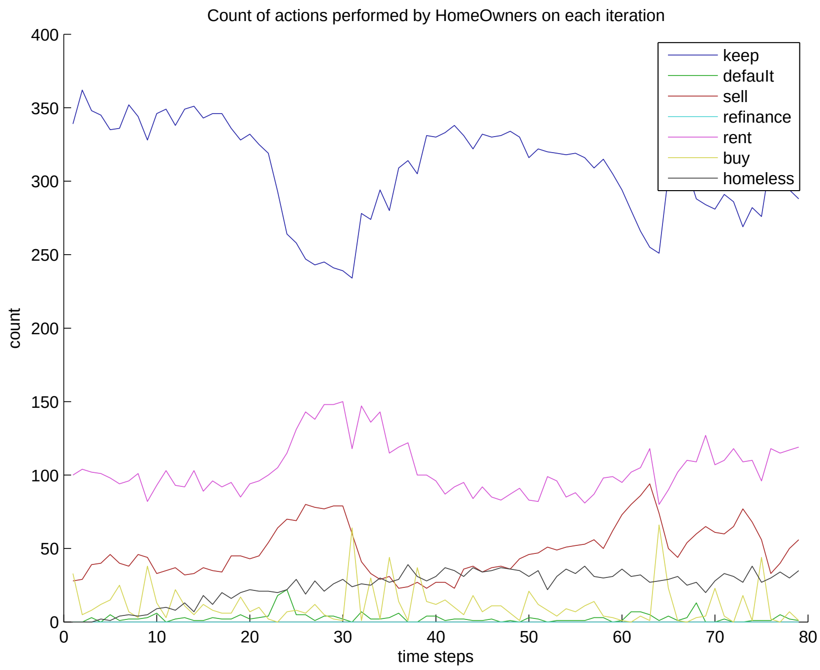

Figure 6: Counts of actions taken from HomeOwners over the course of the simulation from Figure 6.

0.5 Conclusion

In this project we built a model to simulate a very small economic system. We found, as anyonw who runs the code will find, these patterns up ups and downs, that seem to be a common pattern in the real economy. Though the original goal - to simulate the mortgage chrisis - was perhaps a little ambitious for the scope of this project, we did show, with a relatively simple model, patterns of fluctuations that resemble economic bubbles.

The next extension, if this project were to be continued, would be to add investment banks and other parties interested in mortgage-backed securities, as was done in [Goldstein,2011]

Bibliography

[Schelkle,2011] Schelkle T *Simulating Mortgage Defaults* London School of Economics

[Goldstein,2011] Goldstein Jon *An Agent-Based Model of the Interaction between the Housing and RMBS Markets* George Mason Univeristy

[Nersesian,2008] Nersesian Roy *Simulating the Financial Consequences of the Subprime Mortgage Crisis* Monmouth Univeristy

Code

Here for your reading pleasure we include the setup script. It sets initial World values and instantiates objects. For more convenient access, please see the GitHub directory: git@github.com:petered/M.git

```
%==================================================
%                 Input Script
%==================================================

cd(fileparts(mfilename('fullpath')));
close all hidden; clear; clc;

%% Enter Parameters
% Enter initial model parameters here

I=struct;                          % Initialization options

I.S.nSteps=80;                     % Number of Simulation Steps

% World Settings
I.W.houseIndex      = 200;         % Initial house price index
I.W.houseIXhist     = 100;         % Length of house index history to make.
I.W.fedInterest     = .08/12;      % Federal interest rate
I.W.pJobLoss        = .02;         % Probability of job loss
I.W.pJobGain        = .1;          % Probability of getting job
I.W.rent            = .9;          % Rent cost
I.W.pSell           = .03;         % Probability of spontaniously selling a house
I.W.priceMem        = 3;           % Time constant of houseIndex price memory


% Bank Settings
I.B.N               = 6;           % Number of Banks
I.B.liquidMean      = 10000;       % Mean Liquid assets owned by a bank
I.B.liquidShape     = 3;           % Shape of Liquid assets distribution
I.B.mortgageLengths = [15 20 30]*12;% Mortgage length options

% HomeOwner Settings
I.O.N               = 500;         % Number of HomeOwners
I.O.salaryMean      = 12/12;       % Mean salary for housing
I.O.salaryDev       = 2/12;        % Sharpness of salary distribution
I.O.cashMean        = 20;          % Mean salary for housing
I.O.cashDev         = 5;           % Sharpness of cash distribution
I.O.nHousesToSearch = 6;           % Number of houses to search
I.O.salcashCorr     = 0.8;         % Correlation between income and salary
I.O.LTVthresh       = 1.2;         % Loan-To-Value ratio before defaulting
I.O.fJobs           = .9;          % Fraction of people with jobs
I.O.foresight       = 36;          % Months to look in advace when deciding to buy
I.O.panicHorizon    = 12;          % Months to look in advance when deciding if you need to sell.
I.O.discount        = .04/12;      % Discount parameter
I.O.memory          = 25;          % Time constant of homeowner memory
I.O.americanDream   = 20;          % Cash value of the joy of having a house
I.O.saleMarkup      = .1;         % Fraction to mark-up a house when putting it up for sale
I.O.salePanic       = .03;         % How much to decrease the sale price each turn
```

```
% House Settings
I.H.N               = 400;          % Number of houses
I.H.worthMean       = 1;
I.H.worthDev        = .3;           % Deviation of house worth distribution.
I.H.fPeopleOwned    = 0.95;         % Fraction of house owned by people (as opposed to banks)
I.H.salaryCorr      = 0.8;          % Correlation of salary to house price
I.H.fForSale        = .1;           % Fraction of people-owned houses initially for sale

% Mortgage Settings
I.M.fHousesHeld      = 0.7;         % Fraction of houses initially mortgaged (they'll be in random s
I.M.term             = 20*12;          % Mortgage time (should eventually be made a distribution)
I.M.typesProb        = [0.5; 0.5];  % Fraction of existing mortgages that are adjustable.

% Mortgage Type settigns
I.MT(1).adjustable  = false;        %
I.MT(1).down        = 0.2;
I.MT(1).incomeBuf   = 1.2;
I.MT(1).ratePrem    = .02/12;
I.MT(2).adjustable  = true;
I.MT(2).down        = 0.03;
I.MT(2).incomeBuf   = .9;
I.MT(2).teaser      = 0.03*ones(1,12)/12;
I.MT(2).ratePrem    = .02/12; % Surplus rate to fed interest


%% Enter Statistics of interest

frac=@(x)nnz(x)/length(x);

ST=struct;
ST.housingIndex=@(W)W.houseIndex;
ST.fHomesForSale=@(W)frac([W.H.forSale]);

ST.fPeopleWithHouses=@(W)frac(arrayfun(@(x)~isempty(x.H),W.O));
ST.fHousesFree=@(W)frac(arrayfun(@(h)isempty(h.M),[W.H]));
ST.fHousesWithPeople=@(W)frac(arrayfun(@(x)isa(x.Owner,'HomeOwner'),W.H));

ST.fUnemloyment=@(W)frac(~[W.O.hasJob]);
ST.meanCash=@(W)mean([W.O.cash]);
% ST.keepCount=@(W)W.actions.keep;
% ST.sellCount=@(W)W.actions.sell;
% ST.defaultCount=@(W)W.actions.default;
% ST.rentCount=@(W)W.actions.rent;
% ST.buyCount=@(W)W.actions.buy;
ST.salePrice=@(W)mean([W.HforSale.price]);


%% Initialize world
% More advanced tinkering can be done here.
T=struct;       % Structure to store all the temporary initializers.

% A whole new world
W=World;
```

```
W.pJobLoss=I.W.pJobLoss;
W.pJobGain=I.W.pJobGain;
W.fedInterest=I.W.fedInterest;
W.houseIndex=I.W.houseIndex;
W.houseIndexHistory=I.W.houseIndex*ones(1,I.W.houseIXhist);
W.rent=I.W.rent;
W.pSell=I.W.pSell;
W.priceMem=I.W.priceMem;

% Generate Banks
T.Bi=Bank(); T.Bi.initializer;
W.B(I.B.N)=Bank;
W.B.distribute('liquid',gammaDraw(I.B.liquidMean,I.B.liquidShape,I.B.N));
W.B.distribute('Mlen',{I.B.mortgageLengths});

% Generate HomeOwners
W.O(I.O.N)=HomeOwner;
[T.sal T.cash]=skewedDraw([I.O.salaryMean I.O.cashMean],[I.O.salaryDev I.O.cashDev], I.O.N,I.O.salca
W.O.distribute({'salary','cash'},T.sal,T.cash);
W.O.distribute('hasJob',I.O.fJobs>rand(1,I.O.N));
W.O.distribute('nHousesToSearch',I.O.nHousesToSearch);
W.O.distribute('LTVthresh',I.O.LTVthresh);
W.O.distribute('foresight',I.O.foresight);
W.O.distribute('panicHorizon',I.O.panicHorizon);
W.O.distribute('discount',I.O.discount);
W.O.distribute('memory',I.O.memory);
W.O.distribute('americanDream',I.O.americanDream);
W.O.distribute('saleMarkup',I.O.saleMarkup);
W.O.distribute('salePanic',I.O.salePanic);
W.O.distribute('tag',ceil(1000000*rand(1,I.O.N)));
W.O.link('B',W.B,[],true,1,true); % Link each Homeowner to 2 banks
W.O.link('B',W.B,[],true,1,true);

% Generate Houses
W.H(I.H.N)=House;
W.H.distribute('worth',skewedDraw(I.H.worthMean,I.H.worthDev,I.H.N,[],true));
W.H.distribute('price',[W.H.worth]*I.W.houseIndex);

% Generate Mortgage Types
T.MT(length(I.MT)*I.B.N)=MortgageType;
W.MT=T.MT;
[W.MT(1:2:end).adjustable]   = deal(I.MT(1).adjustable);
[W.MT(1:2:end).down]         = deal(I.MT(1).down);
[W.MT(1:2:end).teaser]       = deal(I.MT(1).teaser);
[W.MT(1:2:end).ratePremium]  = deal(I.MT(1).ratePrem);
[W.MT(1:2:end).incomeBuffer] = deal(I.MT(1).incomeBuf);
[W.MT(2:2:end).adjustable]   = deal(I.MT(2).adjustable);
[W.MT(2:2:end).down]         = deal(I.MT(2).down);
[W.MT(2:2:end).teaser]       = deal(I.MT(2).teaser);
[W.MT(2:2:end).ratePremium]  = deal(I.MT(2).ratePrem);
[W.MT(2:2:end).incomeBuffer] = deal(I.MT(1).incomeBuf);
for i=1:length(W.B), link(W.B(i),'MT',W.MT(2*i-1:2*i),'B'); end              % Link mortgage types t
W.MT.distribute('W',W);
```

```
% Link Houses to owners
T.Hlinked=link(W.H,'Owner',W.O,'H',true,I.H.fPeopleOwned);                    % Link of homes to h
% T.Hlinked.correlate('worth',arrayfun(@(h)h.Owner.salary,T.Hlinked),I.H.salaryCorr,true); % Correla
T.Hlinked.distribute('forSale',I.H.fForSale>rand(size(T.Hlinked)));
W.H(arrayfun(@(h)isempty(h.Owner),W.H)).link('Owner',W.B,[],true,1,true);
W.H(arrayfun(@(h)isa(h.Owner,'Bank'),W.H)).distribute('forSale',true);

% Generate Mortgages for some percentage of linking houses
W.M(ceil(length(T.Hlinked)*I.M.fHousesHeld))=Mortgage;
W.M.link('MT',W.MT,[],true,1,true);                                          % Link Mortgages randomly
W.M.link('H',permrand(W.H),'M',true);                                        % Link Mortgages randomly
for i=1:length(W.M),
    W.M(i)=W.M(i).MT.realize(W.M(i).H,I.M.term); W.M(i).H.M=W.M(i);
    W.M(i).startPeriod=ceil(rand(1,length(W.M(i).duration)));
    W.M(i).principal=W.M(i).principal-W.M(i).scheduleP(1:W.M(i).period);
end   % Make Mortgages "fo' real"
W.M.distribute('principal',rand(size(W.M)).*[W.M.principal]);           % Mortgages have paid off s

%% Initialize simulation
S=Simulation;
S.Stats=ST;
W.inTheBeginning;
S.W=W;
S.nSteps=I.S.nSteps;

S.runLink;

%====================================================
```