

## The Programs

The matlab programs have been written to make replication or modification as easy as possible.

The core program is `figs.m`. This program will plot figures and compute the demand for loanable funds for any parameters and functional forms that the investigator may wish to examine.

To search over 1000 data points at a time we have developed a second program `search.m`. Below we describe both programs in detail.

### A. `figs.m`

The core program is `figs.m`. This program will plot figures and compute the demand for loanable funds for any choice of density function for the types  $g(t)$  and density function for the distribution of output about its mean  $mu$ . We define  $x = y - mu$  and a density function for output,  $f(x|t)$ .

Higher types of loan applicant have more risky returns. We model this by introducing a spread variable  $s(t)$ . We assume that this increases linearly with type.

$$s(t) = (1-t)*s_0 + t*s_1$$

We also allow the support of the distribution of output to be higher for higher risk types. We define support parameter for the lowest risk,  $lam_0$  and a spread parameter for the highest risk,  $lam_1$  and define the support parameter for type  $t$  to be a convex combination.

$$lam(t) = (1-t)*lam_0 + t*lam_1$$

The support of  $x = y - mu$  is then  $[-lam(t), lam(t)]$ .

In our research we chose to focus on truncated normal distributions. Types are distributed with (un-normalized) density function

$$g(t|\bar{m}, \bar{s}) = \exp(-0.5*((t - \bar{m})/\bar{s})^2) \text{ over } [0,1]$$

As long as  $\bar{m}$  lies in the support of the distribution it is the mode so we refer to it as the mode parameter. Of course, with no truncation,  $\bar{m}$  is also the mean and  $\bar{s}$  is the standard deviation of the normal distribution.

Similarly the (un-normalized) density function is

$$f(x | t, s(t)) = \exp(-0.5 * (x / s(t))^2), \quad x \in [-lam(t), lam(t)]$$

The program draws figures as output. Figure 1 shows the density function for the types over the support of the type space  $[0,1]$ . Figure 2 shows the density function of the output distribution  $x = y - mu$  for the best and worst risks.

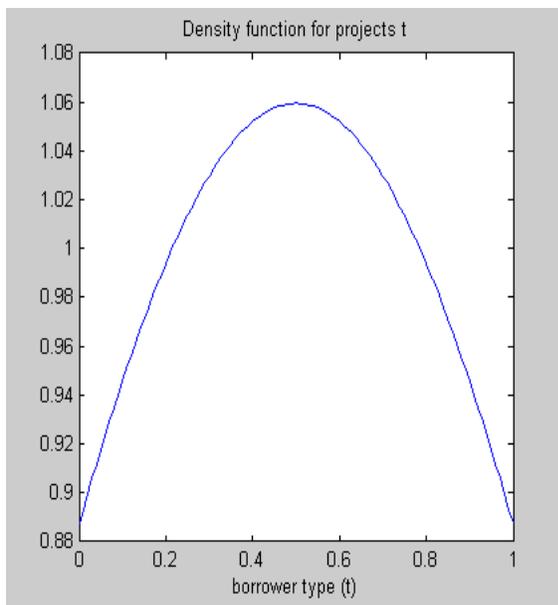


Figure 1

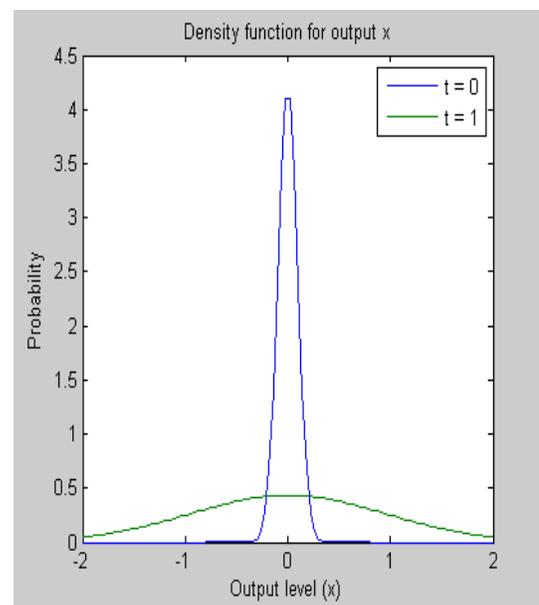


Figure 2

Figure 3 shows the interest rate at which each borrower type exits the market for loans. Figure 4 shows the expected profit for three types as a function of the interest rate.

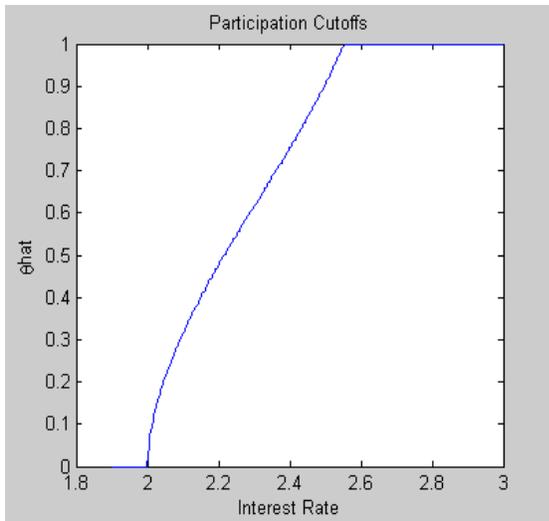


Figure 3

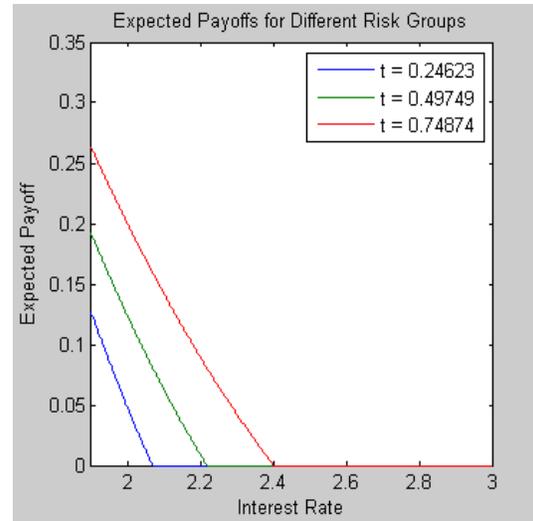


Figure 4

Figure 5 plots the expected profit of borrowers in the market for loans as a function of the interest rate. If there is a local maximum this is marked. The turning point is also part of the program output and appears in the COMMAND WINDOW.

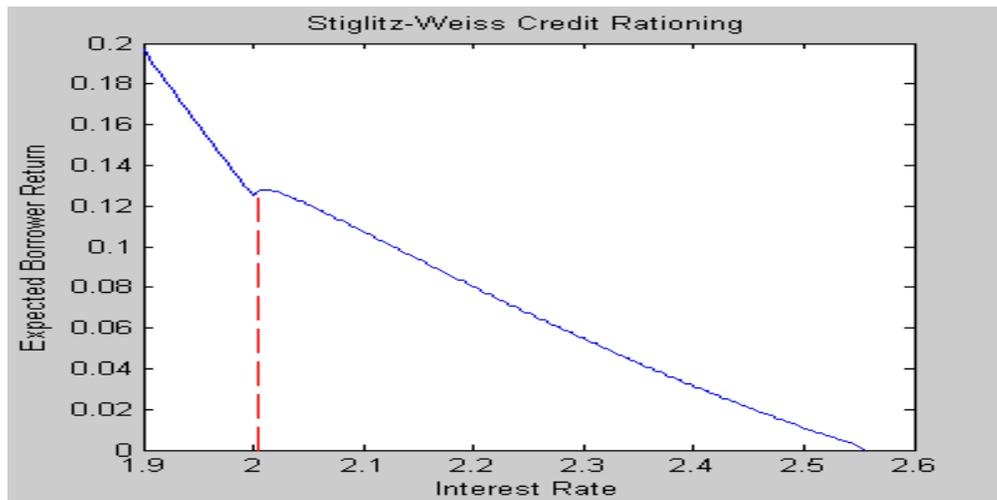


Figure 5

Figure 6 shows the implied demand curve for loanable funds.

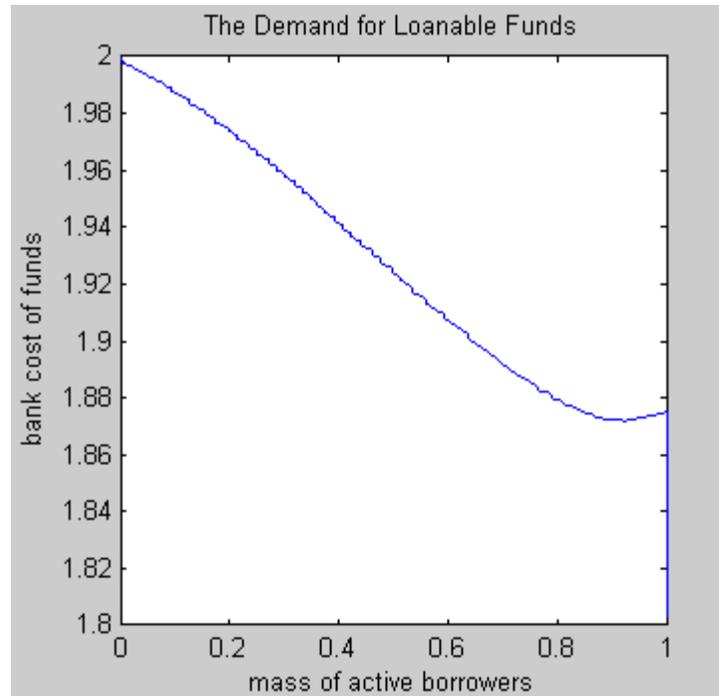


Figure 6

Figure 7 also shows the expected profit of borrowers. It is a smoothed version of Figure 5. This is a relic but is left in for historical reasons.

Finally Figure 8 inverts Figure 5 and shows the expected seller revenue as a function of the interest rate paid by borrowers.

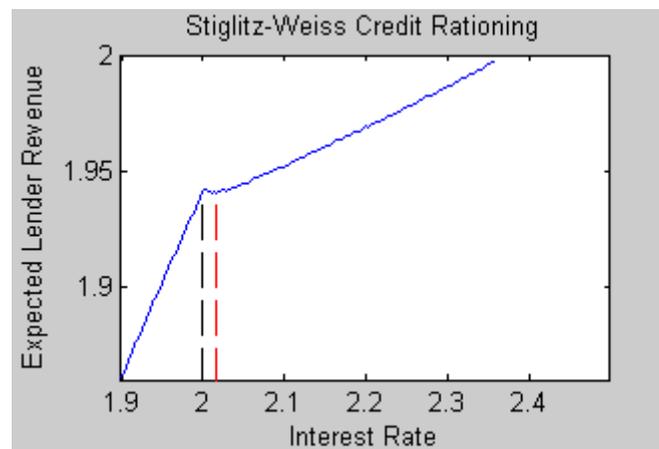


Figure 8: Expected lender revenue

The program also generates output in the COMMAND WINDOW

0:07 22-Feb-2008

## Parameters

'C'	'MU'	'SIGMA0'	'SIGMA1'	'lam0'	'lam1'	'PR{bankr}'	'MUg'	'SIGMAg'
[0.2000]	[ 2]	[0.0953]	[0.9526]	[ 2]	[ 2]	[ 0.0100 ]	[0.5000]	[0.8389]

## PROBABILITY OF BANKRUPTCY

t= 0.00 pb(t)= 0.02

t= 0.50 pb(t)= 0.35

t= 1.00 pb(t)= 0.41

## TAIL PROBABILITIES FOR PROJECT DENSITY

t= 0.50 pr(t)= 0.50

t= 0.70 pr(t)= 0.29

t= 0.90 pr(t)= 0.09

Elapsed time is 147.928828 seconds.

The cutoff interest rate for lending Rstar is: 2.5558

No local minima found

1 local maxima found at r = 2.0048

Mleft = -0.5000

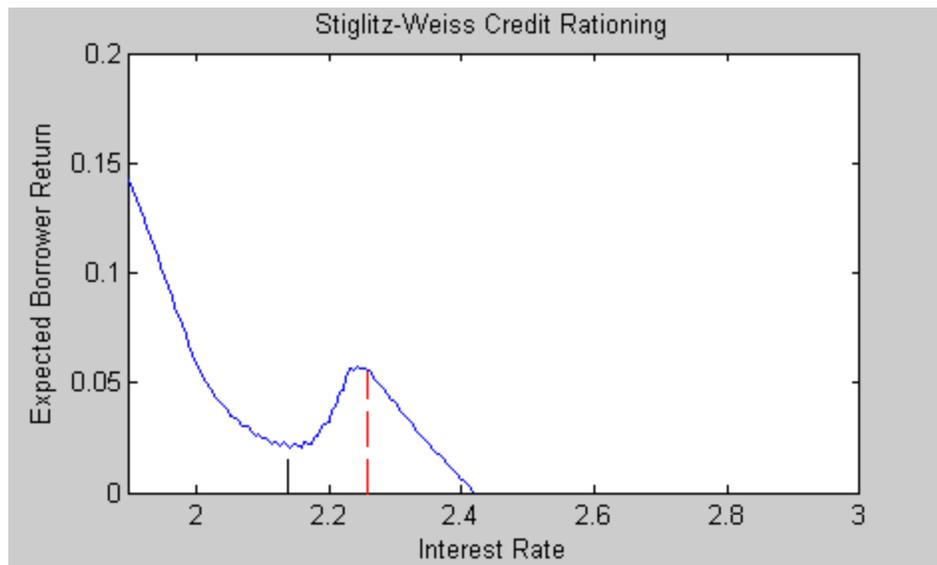
Most importantly, the input values are listed and the first local maximum if any. Note that in this example there is a turning point but the non-monotonicity is small.

As noted in the paper, a sufficient condition for a non-monotonicity is that for some interval of types the density is sufficiently large. However for the normal distribution, the density function can only be made large by simultaneously making the variance of the distribution small. Since it is differences in types that leads to adverse selection, the declining variance offsets the effect of the high density. We can however easily illustrate the “big mode” effect by adding a tail. We

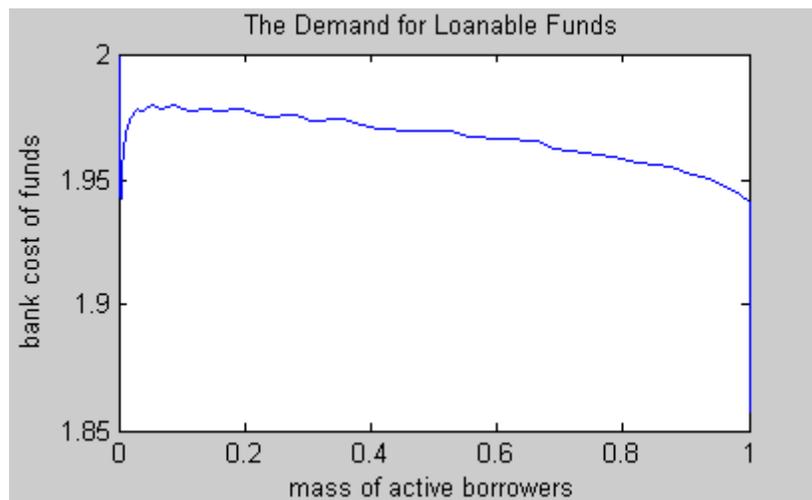
consider a density function which is a convex combination of a normalized truncated normal density function  $n(t, s(t))$  and uniform density function  $u(t)$ , both with support  $[-lam, lam]$ , that is

$$g(t) = (1-U) * n(t, mode, spread) + U * u(t)$$

In the program you can examine this by choosing different values for the parameter  $U$ . The default is  $U = 0$ , that is, the truncated normal distribution. This is illustrated below.

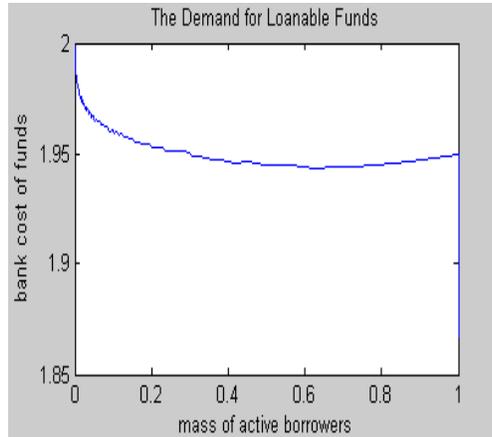
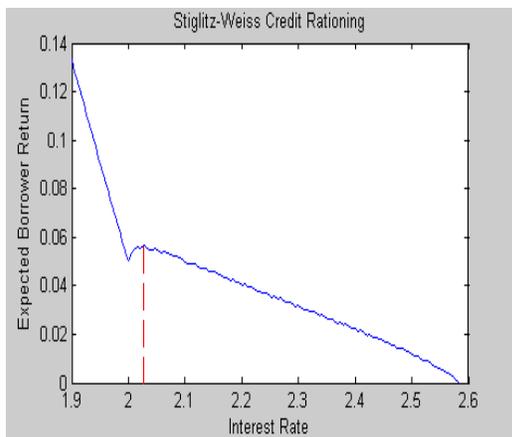
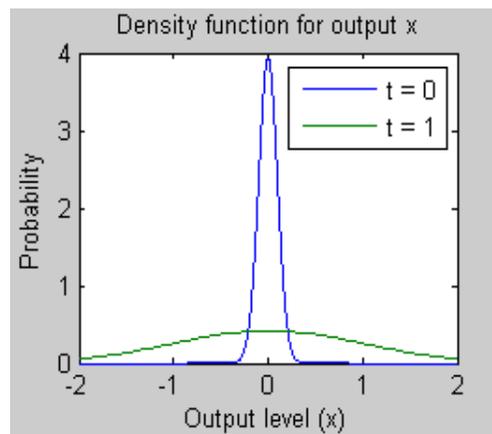
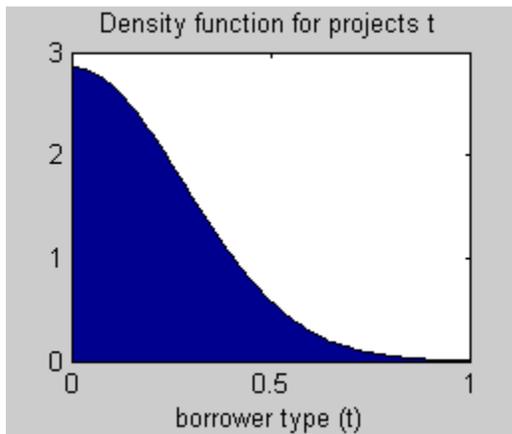


Such examples do not necessarily lead to much rationing as the following diagram indicates.



Using the sister program search.m, we search for turning points over a wide range of parameter values. Based on this search we conclude that when there is such a turning point is one of the extreme cases discussed in the paper. The following example is typical.

C	MU	S0	S1	lam0	lam1	PR{bankr t = 0}	MODE	SPREAD
0.2	2	0.1	1	2	2	0.023	0	0.28



Note in particular that the probability of bankruptcy for the lowest type is small. Note also that the weight in the left tail of the distribution of types is large so that the mass of types exiting at loan rates close to the mean is large.

## B. search.m

This program conducts a search for turning points for 1000 parameter values. The data is read in from a spreadsheet grid.xls . The first set of parameters are the type parameters (MODE,SPREAD). In the worksheet ‘Search’ in you pick 10 values for the MODE parameter and 10 values for  $PR\{tail\} = \text{MIN}(PR\{t \leq 0.1, t \geq 0.9\})$ . The program then calculates the SPREAD parameter for the lowest type that implies a probability tail probability of  $PR\{tail\}$  for each value of MODE. This yields 100 data points.

The screenshot shows the MATLAB 7.4.0 (R2007a) environment. The Editor window displays the script `search002.m` with the following code:

```

27
28 % Read in mode parameters
29 file = 'grid002';
30 sheet = 'sheet1';
31 range = 'b22:k22';
32 [numeric txt raw] = xlsread(file,sheet,range);
33 disp('The 10 values of the MODE parameter for types in the 6 runs');
34 % disp(' 1 2 3 4 5 6 7 8 9 10');
35 disp(raw);
36
37 file = 'grid002';
38 sheet = 'sheet1';
39 range = 'b23:k23';
40 [numeric txt raw] = xlsread(file,sheet,range);
41 disp('The 10 values of PR{tail}=MIN(PR{t<=0.1},PR{t>= 0.9}) in each run');

```

The Command Window shows the output of the script:

```

9:10 27-Feb-2008
The 10 values of the MODE parameter for types in the 6 runs
[0] [0.1000] [0.2000] [0.3000] [0.4000] [0.5000] [0.6000] [0.7000] [0.8000] [0.9000]

-----
RUN #1 of 6
-----
Fixed Parameters
' C' ' MU' ' SIGMA0' ' SIGMA1' ' lam0' ' lam1' ' PR(bankr)' ' run #'
1 2 0.43867 4.3867 2 2 0.01 1
Final Output
MUg SIGMAg #Maxima #Minima 1stMax 1stMin t at 1stMax t at 1stMIN
0 0.3778 0 0 0 0 0 0
0 0.4410 0 0 0 0 0 0
0 0.4993 0 0 0 0 0 0
0 0.5606 0 0 0 0 0 0

```

If you open the program, it will look as above. In this example the 10 values of the mean all lie on the unit interval and appear as output in the Command Window. For each probability of bankruptcy there is a “run” using all 100 data points for the type distribution. Any turning points are identified in the Command Window.

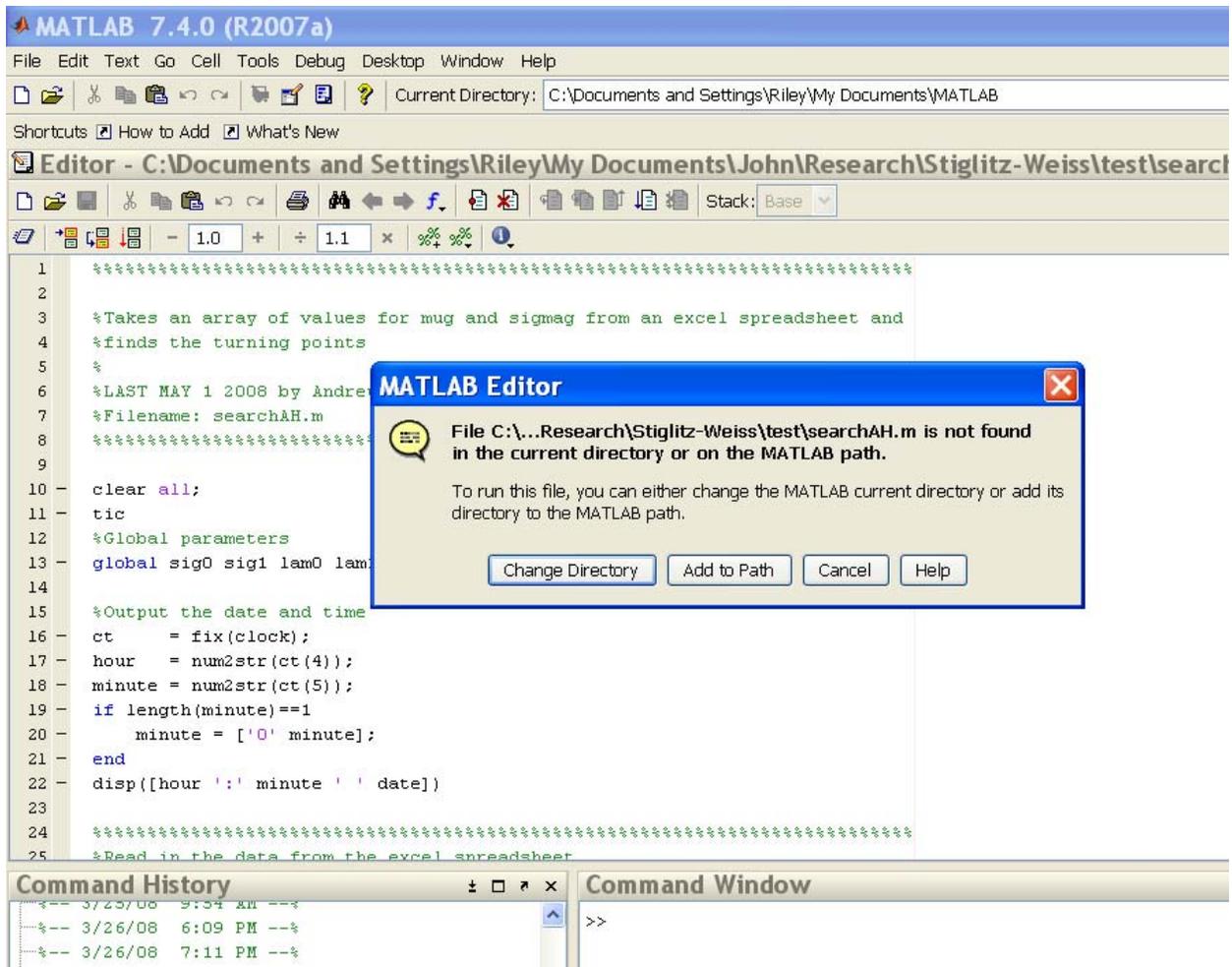
When you change the data in the spreadsheet the spread parameters are all recomputed.

In the worksheet ‘Spread’ also you choose the collateral, the extreme supports for the supports for the output distribution and the extreme spread parameters for each of the ten runs. The program automatically computes the probability of bankruptcy associated with each spread parameter choice. The worksheet is depicted below. The program reads the two input arrays.A5:G12 and B30:C129.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	<b>PARAMETERS FOR SEARCH PROGRAM</b>												
2													
3	<b>Input Array is A5:G12</b>												
4													
5	C	0.2	0.2	0.2	0.2	0.2	0.2	C	collateral				
6	MU	2	2	2	2	2	2	MU	mean output				
7	SIGMA0	0.1	0.2	0.3	0.4	0.5	0.6	SIGMA0	spread paramter for lowest risk type				
8	SIGMA1	2	2	2	2	2	2	SIGMA1	spread paramter for lhighest risk type				
9	lam0	2	2	2	2	2	2	lam0	support parameter for lowest type				
10	lam1	2	2	2	2	2	2	lam1	support parameter for highest type				
11	PR{bk t=0}	0.02275	0.159	0.25	0.31	0.34	0.37	PR{bk t}	probability of backruptcy for type t				
12	run #	1	2	3	4	5	6						
13													
14	PR{bk t=1}	0.44	0.44	0.44	0.44	0.44	0.44						
15													
16													
17	<b>Enter data of the model in the green cells</b>												
18	There are yellow data cells in the 'Types1' and 'Output' worksheets.												
19	However these are technical program parameters rather than model parameters.												
20													
21	<b>Choice of MODES. These are the 100 data points in the Types calcaultions</b>												
22	MODE	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9		
23	PR{tail}	0.001	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09		
24													
25													
26													
27	<b>Input array is B30:C129</b>												
28													
29		MODE	MIN										
30	1	0	0.28										
31	2	0	0.28										

The spread parameters for the distribution of types are all computed from the ten worksheets ‘Types#’. The program is sufficiently robust that it should not be necessary to modify these pages.

To use the program open MATLAB and then the file **search.m**. The program will appear in the Editor window. Click on the down arrow. You will likely be asked to either “Change Directory” or “Add to Path.” Choose the latter option.



The output of a numerical run is a spreadsheet that will be saved in the “Current Directory” (see line three of the screen shot above.) The name of the file is SW[time][date].xls

Col A: Mode of type distribution

Col B: Spread parameter of type distribution

Col C # of maxima of average borrower profit function

Col D # of minima

Col E Interest rate for first local maximum

Col F Interest rate for first local minimum

Col G Type that exits at first maximum

Col H Type that exits at first minimum

Col I Interest rate at first max

Col J Interest rate at first min

Col K Interest rate beyond which average profit is higher than at the first local maximum

Col L Interest rate at which highest risk borrower exits.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1																		
2	C	MU	SIG0	SIG1	lam0	lam1												
3	0.8	2	0.2	2	2	2												
4	MODE	SPREAD	#Maxima	#Minima	1stMax	1stMin	t@1stMax	t@1stMIN	R1	R2	R3	rstar	U1	U2	U3	Pr{ $\theta$ (R2)}	Pr{ $\theta$ (R3)}	
5	-10	2.043571	1	0	2.015789	0	0.185929648	0	2	2.015789	2.079198	2.217969	0.050507	0.06827	0.049797	0.402611503	0.641852255	
6	-10	2.173571	1	0	2.015789	0	0.185929648	0	2	2.015789	2.073684	2.217969	0.054141	0.070871	0.053822	0.376480005	0.600273929	
7	-10	2.308571	1	0	2.013033	0	0.170854271	0	2	2.013033	2.070927	2.217969	0.057437	0.072729	0.055677	0.329476263	0.561380808	
8	-10	2.453571	1	0	2.013033	0	0.170854271	0	2	2.013033	2.065414	2.217969	0.060499	0.074929	0.059585	0.310039825	0.524485578	
9	-10	2.608571	1	0	2.013033	0	0.170854271	0	2	2.013033	2.062657	2.217969	0.063306	0.076915	0.061357	0.292844088	0.489903956	
10	-10	3.016667	1	0	2.010276	0	0.16080402	0	2	2.010276	2.054386	2.217969	0.068924	0.081519	0.067194	0.24610356	0.422039537	
11	-10	2.973571	1	0	2.010276	0	0.16080402	0	2	2.010276	2.054386	2.217969	0.068429	0.081169	0.066976	0.248791809	0.425789186	
12	-10	3.193571	1	0	2.010276	0	0.16080402	0	2	2.010276	2.051629	2.217969	0.070761	0.082815	0.069414	0.236258616	0.402245647	
13	-10	3.458571	1	0	2.010276	0	0.16080402	0	2	2.010276	2.048872	2.217969	0.073025	0.084401	0.071182	0.224413318	0.373737972	
14	-10	3.778571	1	0	2.010276	0	0.16080402	0	2	2.010276	2.046115	2.217969	0.075169	0.085893	0.073621	0.213490962	0.352550448	
15	-9	1.945714	1	0	2.015789	0	0.185929648	0	2	2.015789	2.079198	2.217969	0.050595	0.06831	0.049805	0.401790146	0.641096712	
16	-9	2.070714	1	0	2.015789	0	0.185929648	0	2	2.015789	2.073684	2.217969	0.054255	0.07093	0.053844	0.37552755	0.599302463	
17	-9	2.195714	1	0	2.013033	0	0.170854271	0	2	2.013033	2.070927	2.217969	0.057454	0.072723	0.055663	0.329225238	0.561200023	
18	-9	2.330714	1	0	2.013033	0	0.170854271	0	2	2.013033	2.065414	2.217969	0.060453	0.07488	0.059547	0.310201373	0.524809319	
19	-9	2.480714	1	0	2.013033	0	0.170854271	0	2	2.013033	2.062657	2.217969	0.063311	0.076905	0.061342	0.292700764	0.489809475	
20	-9	2.645714	1	0	2.013033	0	0.170854271	0	2	2.013033	2.057143	2.217969	0.065973	0.078767	0.065217	0.276924239	0.456745172	
21	-9	2.830714	1	0	2.010276	0	0.16080402	0	2	2.010276	2.054386	2.217969	0.068467	0.081186	0.06698	0.248500632	0.42544909	
22	-9	3.040714	1	0	2.010276	0	0.16080402	0	2	2.010276	2.051629	2.217969	0.070799	0.082833	0.069421	0.235983234	0.401912613	
23	-9	3.290714	1	0	2.010276	0	0.16080402	0	2	2.010276	2.048872	2.217969	0.073039	0.084403	0.071179	0.224280551	0.373593027	
24	-9	3.595714	1	0	2.010276	0	0.16080402	0	2	2.010276	2.046115	2.217969	0.075183	0.085897	0.07362	0.213367403	0.352408985	
25	-8	1.842857	1	0	2.015789	0	0.185929648	0	2	2.015789	2.079198	2.217969	0.05071	0.068364	0.049818	0.400734888	0.640115288	
26	-8	1.957857	1	0	2.015789	0	0.185929648	0	2	2.015789	2.073684	2.217969	0.054261	0.070909	0.053817	0.375293014	0.599208215	
27	-8	2.077857	1	0	2.013033	0	0.170854271	0	2	2.013033	2.070927	2.217969	0.057503	0.072736	0.055657	0.328732373	0.560744936	
28	-8	2.207857	1	0	2.013033	0	0.170854271	0	2	2.013033	2.065414	2.217969	0.060545	0.074925	0.059562	0.309473766	0.524007522	
29	-8	2.347857	1	0	2.013033	0	0.170854271	0	2	2.013033	2.062657	2.217969	0.063354	0.076917	0.06134	0.292305258	0.489404728	
30	-8	2.502857	1	0	2.013033	0	0.170854271	0	2	2.013033	2.057143	2.217969	0.065991	0.078765	0.065207	0.276693091	0.456531682	
31	-8	2.677857	1	0	2.010276	0	0.16080402	0	2	2.010276	2.054386	2.217969	0.068482	0.081184	0.066972	0.248311088	0.425265825	
32	-8	2.877857	1	0	2.010276	0	0.16080402	0	2	2.010276	2.051629	2.217969	0.070826	0.082841	0.069419	0.235750809	0.40165228	
33	-8	3.117857	1	0	2.010276	0	0.16080402	0	2	2.010276	2.048872	2.217969	0.07309	0.08443	0.071192	0.223942038	0.373168902	
34	-8	3.402857	1	0	2.010276	0	0.16080402	0	2	2.010276	2.046115	2.217969	0.0752	0.085901	0.073618	0.2132213	0.3522433	

Output of a run