Notes on Computing

Jeremy Greenwood and Nezih Guner*

June 17, 2008

Abstract

These notes discuss several aspects surrounding the algorithm used to compute the model in “Marriage and Divorce since World War II: Analyzing the Role of Technological Progress on the Formation of Household.” The first section explains the basic building blocks of the code that solves the model for the period 1950-2000. A discussion of some additional aspects of the computation follows.

1. The Code

- **Starting the program.** The main code is mainp.m. This code takes a set of parameters, calibrates some household technology parameters using min3prr.m, calculates the 2000 and 1950 steady states using sstate.m, and finally computes the transition between them using trans.m.

- **Calibration.** The parameters whose values are not calibrated by the program are the ones defining tastes, the match quality distribution and household technology. Given these parameters the file min3prr.m calibrates the price decline for household durables, the initial price level of household durables and the fixed cost of household maintenance. These parameters are chosen to minimize the difference between working hours from the data and from the model, as explained in Section 6.1.2 of the paper. In order to minimize this difference min3prr.m uses the function calib3prr.m. The latter also

---

*The authors have made this code available in the interest of economic science. You are free to use it. All that is asked is that you please cite the paper if you do so.
uses the functions hwmarr.m and hwsing.m for finding the solution to the first-order condition of the maximization problem for married and single individuals. After calibrating the parameters, min3prr.m uses their values to compute time paths for effort and utility for the entire period 1950-2000.

- **Computing the Steady States.** The second step is to compute the final and initial steady state. This is done by the code sstate.m. This file is called by mainp.m and it first calculates the value functions as defined in Section 3.1 and 3.2 of the paper. The only input for doing this (defined by mainp.m) is the value of the momentary utility function for the period analyzed, which was already computed by min3prr.m. Therefore the same code sstate.m is used for computing both the initial and the final steady state. After computing the value functions, sstate.m also computes vital statistics using the formulas in Section 4.1 of the paper, obtaining the transition matrix for married agents and the stationary distribution for married and single agents. After estimating the steady states, mainp.m displays the results.

- **Transitional Dynamics.** Finally trans.m is used to calculate the transition between the two steady states, which is used for Figures 6 and 7 in the paper. Using the time path for momentary utility and the value functions for the final steady state it computes time series for matching thresholds backwards starting from the final steady state. After doing this it gets the value functions for every year. With these value functions the code computes the distribution for married agents for every year moving forward in time from the initial steady state. Once this is computed other vital statistics can be estimated for the transition, including the proportion of married agents, the rate of marriage and divorce and the duration of marriage and single states. The code ends plotting the results.

2. Some Computation Issues

2.1. Computing conditional means with a normal pdf

Let $\Phi$ and $\varphi$ be the cumulative density and the probability density functions for the standard normal distribution. First note that
\[ \int_a^b x \varphi(x) \, dx = \int_a^b x \frac{1}{(2\pi)^{1/2}} \exp \left( -\frac{(x)^2}{2} \right) \, dx = \frac{1}{\sqrt{2\pi}} \left( e^{-\frac{1}{2}a^2} - e^{-\frac{1}{2}b^2} \right) = \varphi(a) - \varphi(b). \]

Now let \( N \) be a normal distribution with mean \( \mu \) and standard deviation \( \sigma \). Then, integration by substitution, with the transformed variable \( z = (x - \mu)/\sigma \), yields

\[ \int_a^b x N(x) \, dx = \int_a^b x \frac{1}{\sigma(2\pi)^{1/2}} \exp \left( -\frac{(x - \mu)^2}{2\sigma^2} \right) \, dx = \mu \left[ \Phi \left( \frac{b - \mu}{\sigma} \right) - \Phi \left( \frac{a - \mu}{\sigma} \right) \right] + \sigma \left[ \varphi \left( \frac{a - \mu}{\sigma} \right) - \varphi \left( \frac{b - \mu}{\sigma} \right) \right]. \]

### 2.2. Computing the expected value of a linearly interpolated value function

Now, the problem at hand is to compute

\[ \int_a^b V(x) N(x) \, dx, \]

where \( V(x) \) is the value function and again \( N \) is some normal distribution. Suppose one knows \( V(a) \) and \( V(b) \) and is willing to take a linear interpolation between the points \( a \) and \( b \). Then, for \( a < x < b \)

\[ \int_a^b V(x) N(x) \, dx = \int_a^b [(1 - \chi)V(a) + \chi V(b)] N(x) \, dx, \]

where \( \chi = (x-a)/(b-a) \). The above formula can thus be rewritten as

\[ \int_a^b V(x) N(x) \, dx = \int_a^b \frac{[(b-x)V(a) + (x-a)V(b)]}{b-a} N(x) \, dx \]

\[ = \int_a^b \frac{bV(a) - aV(b) + x[V(b) - V(a)]}{b-a} N(x) \, dx \]

\[ = \frac{[V(b) - V(a)]}{b-a} \int_a^b x N(x) \, dx + \frac{bV(a) - aV(b)}{b-a} \left[ \Phi \left( \frac{b - \mu}{\sigma} \right) - \Phi \left( \frac{a - \mu}{\sigma} \right) \right]. \]
Note that $\int_a^b xN(x)dx$ can be computed using the earlier formula. Also observe that for $t \in [a,b]$

$$
\int_t^b V(x)N(x)dx = \int_t^b \left[\frac{(b-x)V(a) + (x-a)V(b)}{b-a}\right]N(x)dx \\
= \int_t^b \frac{bV(a) - aV(b) + x[V(b) - V(a)]}{b-a}N(x)dx \\
= \frac{[V(b) - V(a)]}{b-a} \int_t^b xN(x)dx + \frac{bV(a) - aV(b)}{b-a} \left[\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{t-\mu}{\sigma}\right)\right].
$$

Analogously, imagine computing the integral on $[b,\infty)$ by extending the linear interpolation on $[a,b]$ beyond $b$.

$$
\int_b^\infty V(x)N(x)dx = \int_b^\infty \left[\frac{(b-x)V(a) + (x-a)V(b)}{b-a}\right]N(x)dx \\
= \int_b^\infty \frac{bV(a) - aV(b) + x[V(b) - V(a)]}{b-a}N(x)dx \\
= \frac{[V(b) - V(a)]}{b-a} \int_b^\infty xN(x)dx + \frac{bV(a) - aV(b)}{b-a} \left[1 - \Phi\left(\frac{b-\mu}{\sigma}\right)\right].
$$

2.3. Computing the threshold value

Suppose that the threshold value, $t$, is known to lie within the two adjacent grid points $a < b$. Then the threshold value, $t \in [a,b]$, must solve the equation

$$V(t) = W,$$

so that

$$\frac{bV(a) - aV(b) + t[V(b) - V(a)]}{b-a} = W,$$

or

$$t = \frac{b-a}{[V(b) - V(a)]}W + \frac{aV(b) - bV(a)}{[V(b) - V(a)]}.$$

2.4. Deriving the normal pdf and cdf using the error function

Finally, let erf be the error function. It is defined by

$$
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.
$$
Since
\[ \varphi(t) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-t^2}{2}\right), \]
we have
\[ \varphi(\sqrt{2}t) = \frac{1}{\sqrt{2\pi}} \exp(-t^2). \]
Hence,
\[ \text{erf}(x) = 2 \int_0^x e^{-t^2} dt = 2 \int_0^x e^{-t^2} dt = 2\sqrt{2} \int_0^x \varphi(\sqrt{2}t) dt. \]
Let \( s = \sqrt{2}t \) and use change of variables to get
\[ \text{erf}(x) = 2 \int_0^{\sqrt{2}x} \varphi(s) ds = 2 \left[ \Phi(\sqrt{2}x) - \frac{1}{2} \right]. \]
Then,
\[ \text{erf}\left(\frac{x}{\sqrt{2}}\right) = 2\Phi(x) - 1, \]
or
\[ \Phi(x) = \frac{\text{erf}\left(\frac{x}{\sqrt{2}}\right) + 1}{2} \text{ if } x > 0, \]
and
\[ \Phi(x) = 1 - \Phi(-x) \text{ if } x < 0, \]
where the last equation follows from the fact that the error function is symmetric.

2.5. Using the error function in Matlab

Observe the following:

1. If \( b - \mu > a - \mu > 0 \) then
\[ \Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right) = \frac{\text{erf}\left(\frac{b-\mu}{\sqrt{2}\sigma}\right)}{2} - \frac{\text{erf}\left(\frac{a-\mu}{\sqrt{2}\sigma}\right)}{2} \]
\[ = \frac{\text{merdf}\left(\frac{b-\mu}{\sqrt{2}\sigma}\right) - \text{merdf}\left(\frac{a-\mu}{\sqrt{2}\sigma}\right)}{2}, \]
where \text{merf} is the Matlab error function.
2. If \( a - \mu < b - \mu < 0 \) then

\[
\Phi \left( \frac{b - \mu}{\sigma} \right) - \Phi \left( \frac{a - \mu}{\sigma} \right) = \frac{\text{erf} \left( \frac{-a - \mu}{\sqrt{2} \sigma} \right)}{2} - \frac{\text{erf} \left( \frac{-b - \mu}{\sqrt{2} \sigma} \right)}{2}
\]

\[
= \left[ \text{merf} \left( \frac{b - \mu}{\sqrt{2} \sigma} \right) - \text{merf} \left( \frac{a - \mu}{\sqrt{2} \sigma} \right) \right]/2,
\]

since for the Matlab error function

\[-\text{merf}(x) = \text{merf}(-x), \text{ for } x < 0.\]

3. If \( a - \mu < 0 < b - \mu \) then

\[
\Phi \left( \frac{b - \mu}{\sigma} \right) - \Phi \left( \frac{a - \mu}{\sigma} \right) = \frac{\text{erf} \left( \frac{b - \mu}{\sqrt{2} \sigma} \right)}{2} + \frac{\text{erf} \left( \frac{-a - \mu}{\sqrt{2} \sigma} \right)}{2}
\]

\[
= \left[ \text{merf} \left( \frac{b - \mu}{\sqrt{2} \sigma} \right) - \text{merf} \left( \frac{a - \mu}{\sqrt{2} \sigma} \right) \right]/2,
\]

since again for the Matlab error function

\[-\text{merf}(x) = \text{merf}(-x), \text{ for } x < 0.\]

2.6. Allowing for Autocorrelation in \( M \)

Here is a way to model the autocorrelated process so that long-run distribution of \( b \) has a constant variance and a mean different than zero.

\[b' = (1 - \rho)\mu + \rho b + \sigma \sqrt{1 - \rho^2} \xi, \text{ with } \xi \sim N(0, 1).\]

Then,

\[E(b) = \mu \text{ and } Var(b) = \sigma^2.\]

Note also that in this case

\[b' \sim N \left( (1 - \rho)\mu + \rho b, \sigma^2 \left( 1 - \rho^2 \right) \right),\]

and

\[
\frac{b' - (1 - \rho)\mu - \rho b}{\sigma \sqrt{1 - \rho^2}}
\]

is a standard normal.
References

% mainp.m
% ______________________________________________ MAIN PROGRAM ______________________________________________


% The authors have made this code available in the interest of economic science. You are free to use it. If you do so the authors would be grateful if you cite the paper. This program runs under MATLAB 7.4. Some notes are contained in the readme documentation.

clear all
clear global
close all

global cbar rho kap alph zeta phi price theta pstart pdec ...
wstart winc wend wage sdate edate wgts datatar

% ____________________________________________PARAMETER VALUES ____________________________________________

% start and end dates
sdate = 1950;
edate = 2000;
dur = edate-sdate;
years = linspace(sdate,edate,dur+1);

% TASTES
betal = 0.96; % Discount factor
lifeexp = 47;
delta = 1/lifeexp;
beta = betal*(1-delta);

lspan = 60; % Lifespan -- if used.

alph = 0.278; % Weight on market goods in utility
zeta = -1.901; % Curvature parameter for home goods

% HOUSEHOLD TECHNOLOGY

theta = 0.206; % Weight on goods in household technology.
kap = 0.189; % Elasticity of substitution between time and good in home production
% theta and kap come from McGrattan et al (1997)

phi = 0.7655; % Scale Economies in Household Production
% OECD scale, 2^phi=1+0.7
% STOCHASTIC STRUCTURE  
% match quality parameters

meanmx = 0.5211;         % mean for married distribution
varm = 0.6801;         % variance for married distribution
sdmx = sqrt(varm);
means = -4.252;         % mean for singles distribution
vars = 8.0625;         % variance for singles distribution
sds = sqrt(vars);         % S.d. of Shock for a Single Agent
rho = 0.896;          % autocorrelation for married distribution
sdm = sdmx*( (1-rho^2)^0.5);  % Adjusting std deviation for autocorr
varm = sdm^2;
meanm = (1-rho)*meanmx;         % Adjusting mean for autocorrelation

% LOAD DATA USED IN SIMULATION

load wagedata5020.txt          % Wage data GDP per hours worked, NIPA
wagedata = wagedata5020/100;   % 1950=1
wstart   = wagedata(1);        % start and end of wage data
wend     = 2.99;               % wagedata(n1);

%__________________________START CALIBRATION PROGRAM__________________

min3prr  % m-file containing calibration program
% min3prr calibrates cbar, pstart (i.e. p1950), pdec

%__________________________COMPUTE FINAL STEADY STATE_________________

% CONSTRUCT GRID -- for value functions

center = (meanm+means)/2;
span = 8*(max(sds, sdm));
mqlow   = center - span/2;  % Low value for match quality shock
mqhigh  = center + span/2;  % High value for match quality shock
ngrid   = 400;              % Number of grid points
grid    = [mqlow: (span)/(ngrid-1): mqhigh]'; % Construct grid

% COMPUTE CDF ALONG GRID POINTS

cdfms = sncdf( (grid - means)/sds )';  % Cdf on grid for singles--row
vec

% INITIALIZE VALUE FUNCTIONS

Vj      = zeros(ngrid,1);   % Value func on jth iteration -- married
Vjp1    = Vj;               % Value func on j+1 th iteration -- married
Wj  = 0;                % Value func on jth iteration -- single
Wjp1 = Wj;               % Value func on j+1 th iteration -- single

% PAYOFFS FROM MARRIED AND SINGLE LIFE

um = max(umvec);               % Married utility -- final steady state
us = max(usvec);               % Single utility -- final steady state

% COMPUTE FINAL STEADY STATE

disp(' ')                    % DISPLAY RESULTS
COMPUTING FINAL STEADY STATE--takes time')
sstate                        % m-file containing steady-state program

% DISPLAY RESULTS

disp(' FINAL STEADY STATE')
disp(edate)
disp('Married, Single')
disp([m 1-m])

    disp('Remaining Married, Divorcing, Death=Becoming Single')
disp([prmar*(1-delta)*m (1-delta)*m*(1-prmar) delta*m])

    disp('Marrying, Remaining Single, Death=Remaining Single')
disp([(1-delta)*(1-m)*pnmar (1-delta)*(1-m)*(1-pnmar) delta*(1-m)])

disp('Divorce Rate and Marriage Rate')
disp([1-prmar pnmar])

    disp('check steady state 1')
disp('remain married + marrying + delta*pnmar = m')
disp([prmar*(1-delta)*m+(1-delta)*(1-m)*pnmar+delta*pnmar m])

    disp('check steady state 2')
disp('divorced + remain single + delta*(1-pnmar) = 1-m')
disp([(1-delta)*m*(1-prmar)+(1-delta)*(1-m)*(1-pnmar)+delta*(1-pnmar) l-m])

    dm = 1/(1-prmar*(1-delta));
    ds = 1/(1-(1-pnmar)*(1-delta));

    fm=dm/(dm+ds);

disp('duration of marriages')
disp([dm])
disp('duration of singlehood')
disp([ds])
disp('fraction of time spent married')
disp([fm])
plot(mugrid,muprob)
title('Distribution over Match Quality for Married Agents')
xlabel('Match Quality')
ylabel('Number of Married Agents')
pause(1)
close

disp('PRESS ANY KEY TO CONTINUE')
disp('================================')
pause

% SAVE RESULTS

Wjn=Wj;
Vjn=Vj;
thresn=thres;
muprobn = muprob;

% __________________________COMPUTE INITIAL STEADY STATE________________________

% INITIALIZE VALUE FUNCTIONS

Vj      = zeros(ngrid,1);     % Value func on jth iteration -- married
Vjp1    = Vj;                 % Value func on j+1 th iteration--married
Wj      = 0;                  % Value func on jth iteration -- single
Wjp1    = Wj;                 % Value func on j+1 th iteration--single

% PAYOFFS FROM MARRIED AND SINGLE LIFE

um = min(umvec);              % Married utility -- initial steady state
us = min(usvec);              % Single utility -- initial steady state

% COMPUTING INITIAL STEADY STATE

disp('')
disp(' ')               COMPUTING INITIAL STEADY STATE--takes time')
sstate                % m-file containing steady-state program

% DISPLAY RESULTS

disp('') INITIAL STEADY STATE')
disp(sdate)
disp('Married, Single')
disp([m 1-m])

disp('Remaining Married, Divorcing, Death=Becoming Single')
disp([prmar*(1-delta)*m (1-delta)*m*(1-prmar) delta*m])

disp('Marrying, Remaining Single, Death=Remaining Single')
disp([(1-delta)*(1-m)*pnmar (1-delta)*(1-m)*(1-pnmar) delta*(1-m)])
Divorce Rate and Marriage Rate
\[ \text{check steady state 1} \]
\[ \text{remain married + marrying + delta*pnmar = m} \]
\[ \text{check steady state 2} \]
\[ \text{divorced + remain single + delta*(1-pnmar) = 1-m} \]
\[ \text{duration of marriages} \]
\[ \text{duration of singlehood} \]
\[ \text{fraction of time spent married} \]

\[
d_m = \frac{1}{(1-prmar)(1-delta)}; \\
d_s = \frac{1}{1-(1-pnmar)(1-delta)}; \\
f_m = \frac{dm}{dm+ds};
\]

% SAVE RESULTS
muprobo=muprob;

% __________________________COMPUTE TRANSITIONAL DYNAMICS______________________
trans % m-file for transitional dynamics
% Calibration of pdec, pstart, cbar
% Computation of time paths for utility

% LOAD DATA (TARGET)
% hours data from IPUMS
% this is the data in Figures 3 and 6 in the paper

hrs2454= [1950 0.256 0.313;
          1960 0.268 0.296;
          1970 0.28 0.287;
          1980 0.302 0.296;
          1990 0.335 0.307];

% hrs2454 report "weekly hours/100"
% so we multiply it by 100 and devide by 112 (available time in a week)
% to find the fraction of time spent in the market

hrs2454=[hrs2454(:,1) hrs2454(:,2:3).*(100/112)];

% Weight target by numbers in married and single categories
wgts = [0.82 0.83 0.80 0.71 0.66];

% MINIMIZATION

datatar = hrs2454(:,2:3);   % target

guess(1) = 0.0862   ;    % pdec
guess(2) = 40 ;          % pstart
guess(3) = 0.1;          % cbar

options=optimset('display','iter');
fminsearch('calib3prr', guess, options)
disp('price decline, 1950 price, fixed cost')
disp('==================================')

disp('PRESS ANY KEY TO CONTINUE')
pause

% COMPUTE TIME PATHS FOR EFFORT AND UTILITY

pvec = pstart*exp( (-pdec)*((1:dur+1)-1) );
 % prices used for computing time paths
wvec = wstart*exp( (winc)*((1:dur+1)-1) );
 % wages used for computing time paths
options = optimset('TolFun', 1e-12,'TolX', 1e-12,'disp','none');

for i=1:dur+1;

    price=pvec(i);
    wage=wvec(i);

    J = (( (1-theta)/theta)*price)^( 1/(kap-1) );  % Useful cons
    hwork = (1-alph)*(1-theta)*2 ...
        / ( alph*(theta*J^kap+(1-theta))+(1-alph)*(1-theta)*(1+price*J) );

    hwmvec(i) = fzero('hwmarr',hwork,options);  % Time path for married hwork
    hwsvec(i) = fzero('hwsing',0.9,options);  % Time path for single hwork

    hgmvec(i) = J*(hwmvec(i));  % Time path for marr hld goods
    hpmvec(i) = ( theta*hgmvec(i)^kap + (1-theta)*(hwmvec(i))^kap )^(1/kap);  % Time path for marr hld prod

    cmvec(i) = wage*(2 - hwmvec(i) - price*(hgmvec(i)) );  % Time path for married cons

    umvec(i) = alph*log( (cmvec(i)-cbar)/2^phi ) ...  % Time path for married utility
        + (1-alph)*(hpmvec(i)/2^phi)^zeta/zeta;

    hgsvec(i) = J*hwsvec(i);  % Time path for sin hld goods
    hpsvec(i) = ( theta*hgsvec(i)^kap + (1-theta)*(hwsvec(i))^kap )^(1/kap);  % Time path for sin hld prod

    csvec(i) = wage*(1 - hwsvec(i) - price*(hgsvec(i)) );  % Time path for sin cons

    usvec(i) = alph*log( csvec(i) - cbar ) ...  % Time path for single utility
        + (1-alph)*(hpsvec(i))^zeta/zeta;

    duvec(i) = umvec(i) - usvec(i);  % Time path for util difference

    fmmvec(i) = (2-hwmvec(i))/2;  % Market work -- marr females
    kmsvec(i) = 1-hwsvec(i);  % Market work -- sing females

end

figure(1)
plot(hrs2454(:,1), hrs2454(:,2), years(1:41), fmmvec(1:41))
title('Female labor-force particpation, married')
ylabel('percentage')
disp('================================')
disp('FIGURE 5 in the paper, MARRIED')
disp('PRESS ANY KEY TO CONTINUE')
disp('================================')
pause
close
figure(2)
plot(hrs2454(:,1), hrs2454(:,3), years(1:41), fmseve(1:41))
title('Female labor-force participation, single: data and model')
disp('================================')
disp('FIGURE 5 in the paper, SINGLES')
disp('PRESS ANY KEY TO CONTINUE')
disp('================================')
pause
close
% calib3prr.m
% This function is used to calibrate the initial price of household
% products, pstart, its rate of decline, pdec, and the fixed cost of
% household maintainance, cbar

function criteria = calib3prr(input)

global cbar kap alph zeta phi price theta pstart  pdec   ...
   winc wstart wend wage sdate edate wgts  datatar

pdec   = input(1);
pstart  = input(2);
cbar    = input(3);

% TARGETS

target = [datatar(:,1)' datatar(:,2)'];

edatet = 1990;              % the calibration is only until 1990
dur   = edatet-sdate;
tspan = length(target)/2-1;

%_________COMPUTE TIME PATHS FOR EFFORT AND UTILITY__________________%

pvec = pstart*exp( (-pdec*dur/tspan)*((1:tspan+1)-1) );  % prices given pdec, pstart
winc = log(wend/wstart)/(2000-1950);
wvec = wstart*exp( (winc*dur/tspan)*((1:tspan+1)-1) );     % smooth wage series

options  = optimset('TolFun', 1e-12,'TolX', 1e-12,'disp','none');

for i=1:tspan+1;
    price=pvec(i);
    wage=wvec(i);

    J  = ( ( (1-theta)/theta )*price )^( 1/(kap-1) );  % Useful const
    hwork = (1-alph)*(1-theta)*2 ...%\frac{1}{\theta} \left(\frac{1}{\theta} J + (1-\theta) \right)
        / ( alph*(theta*J^kap+(1-theta))+(1-alph)*(1-
        theta)*(1+price*J) );
    hwmvec(i) = fzero('hwmarr',hwork,options);       % Time path for marr housework
    hwsvec(i) = fzero('hwsing',0.5,options);          % Time path for sin housework

    hgmvec(i) = J*(hwmvec(i));
% Time path for marr hld goods
hpmvec(i) = ( theta*hgmvec(i)^kap + (1-theta)*(hwmvec(i))^kap )^(1/kap);

% Time path for marr hld prod
fmmvec(i) = (2-hwmvec(i))/2;     % Market work -- marr females
fmsvec(i) = 1-hwsvec(i);         % Market work -- sin females

end

flip = [fmmvec, fmsvec];
criteria = 100*sqrt( norm( wgts.*(log(datatar(:,1)') - log(fmmvec)) ) +
                      ... norm( (1-wgts).*{ log(datatar(:,2)') - log(fmsvec) } ) )
/length(fmmvec);
% hwmarr.m
% [zero] = hwmarr(guess)
% hwm = the level of housework for a married couple
% This m-file solves for the married level of housework

function zero = hwmarr(guess)
global cbar kap alph zeta phi price theta wage

hwm   = min(2, max(0,guess) );

J = ( ( (1-theta)/theta )*price )^( 1/(kap-1) ); % A useful constant

% First-order condition for housework -- married

zero = alph*( theta*J^kap+(1-theta) )^(1-zeta/kap)*hwm^(1-zeta) ... 
     - (1-alph)*(1-theta)*2^(-phi*zeta)*( 2 - cbar/wage - hwm - 
       price*J*hwm);

% hwsing.m
% [zero] = hwsing(guess)
% hws = the level of housework for a single couple
% This m-file solves for the single level of housework

function zero = hwsing(guess)
global cbar kap alph zeta phi price theta wage

hws   = min(1, max(0,guess) );

J     = ( ( (1-theta)/theta )*price )^( 1/(kap-1) ); % A useful constant

% First-order condition for housework -- single

zero = alph*( theta*J^kap+(1-theta) )^(1-zeta/kap)*hws^(1-zeta) ... 
     - (1-alph)*(1-theta)*1^(-phi*zeta)*( 1 - cbar/wage - hws - 
       price*J*hws );
% nmean.m
function cmeans = nmean(lower, upper, shock, mean, var)
global rho

% This m-file computes a vector of conditional means for a stand normal pdf
% cmeans = nmean(lower, upper, shock, mean, var)
% lower    = vector of lower bounds
% upper    = vector of upper bounds
% mean     = mean of snpdf
% var      = variance of snpdf
% cmeans   = vector of conditional means
% shock    = current value of shock -- should be zero for iid case

meansp = mean + rho*shock; % Cond mean of shock for next period -- AR1

sd = sqrt(var); % Compute standard deviation.

X = (1/(sqrt(2*pi)))*exp((-((lower-meansp)/sd).^2) / 2 );
Y = (1/(sqrt(2*pi)))*exp((-((upper-meansp)/sd).^2) / 2 );

Z = (erf( ((upper-meansp)/sd)./sqrt(2))/2) ... 
   -(erf(((lower-meansp)/sd)./sqrt(2))/2);

cmeans = (meansp*( Z ) +  sd*( X - Y ))';

% sncdf.m
function cdf = sncdf(x)

% Returns the values for a stand normal cdf given a vector of values, x

% Remove the negative components from x.
posx = max(0,x);
pos  = ( erf(posx/sqrt(2))+1 )/2;
% Note zeros will return a value of 1/2
% Remove the positive components from x.
negx = min(0,x);
neg  = 1-(erf(-negx/sqrt(2))+1)/2;
% Note zeros will return a value of 1/2
cdf = neg + pos -.5; % Subt 1/2 to control for zero values.
% This m file computes a steady state for the model
% Computes value functions and stationary distributions

% START ITERATION OVER VALUE FUNCTIONS

while dist >= .00001 && rlspan<=300   % Tolerance for decision rule

[vptr, nptr] = max( min(0,grid-thres) ); % Nearest right grid pt to "thres"
nptr = nptr-1;   % Nearest left grid pt to "thres"
lower = [thres, grid(nptr+1:ngrid-1)']; % Column vec of lower bounds
upper = grid(nptr+1:ngrid);   % Column vec of upper bounds
condms = nmean(lower,upper,0,means,vars);
% Row vec of cond means -- sing

diffvj = Vj(nptr+1:ngrid)   - Vj(nptr:ngrid-1);   % First diffs of Vj
diffg = grid(nptr+1:ngrid) - grid(nptr:ngrid-1); % First diffs of grid
dervj = diffvj./diffg;                 % Slope of v func for marrieds
intcpt = ( Vj(nptr:ngrid-1).*grid(nptr+1:ngrid) ... - Vj(nptr+1:ngrid).*grid(nptr:ngrid-1) )./diffg;
dcdfs = cdfms(nptr+1:ngrid) ... 
     [sncdf( (thres-means)/sds ), cdfms(nptr+1:ngrid-1)];    % First diff of single CDF
rtailsm = nmean(grid(ngrid),1000,0,means,vars);

for i = 1:ngrid;

cdfmm = sncdf( (grid-rho*grid(i)-meanm)/sdm )';

% Cdf on grid for married agents
dcdfm = cdfmm(nptr+1:ngrid)- ... 
     [sncdf((thres-rho*grid(i)-meanm)/sdm), cdfmm(nptr+1:ngrid-1)];  % First diff of married CDF
condmm = nmean(lower,upper,grid(i),meanm,varm);
% Row vec of cond means -- married
rtailmm = nmean(grid(ngrid),1000,grid(i),meanm,varm);    % Cond mean for right tail -- married

Vjpl(i) = um + grid(i) + beta*sncdf((thres-rho*grid(i)-meanm)/sdm)*Wj ... + beta*dcdfm*intcpt + beta*condmm*dervj ...
+ beta*( ( Vj(ngrid)-Vj(ngrid-1) )/(grid(ngrid)-grid(ngrid-1))*rtailmm ...
\[\begin{align*}
&+ ((\text{grid(ngrid)}*Vj(ngrid-1) - \text{grid(ngrid-1)}*Vj(ngrid))) \ldots \\
&/(\text{grid(ngrid)} - \text{grid(ngrid-1)})) \ldots \\
&*(1 - \text{sncdf}( (\text{grid(ngrid)} - \rho*\text{grid(i)} - \text{meanm})/\text{sdm} )) \ldots \\
&\text{end}
\end{align*}\]
\[\begin{align*}
\text{Wjp1} &= \text{us} + \beta*\text{sncdf}( (\text{thres} - \text{means})/\text{sds} )*\text{Wj} \ldots \\
&+ \beta*\text{dcdfs*intcpt} + \beta*\text{condms*dervj} \ldots \\
&+ \beta* ((\text{Vj(ngrid)} - \text{Vj(ngrid-1)})/(\text{grid(ngrid)} - \text{grid(ngrid-1)}))*\text{rtails} \\
&+ ((\text{grid(ngrid)}*\text{Vj(ngrid-1)} - \text{grid(ngrid-1)}*\text{Vj(ngrid)})/\text{grid(ngrid)} - \text{grid(ngrid-1)})) \ldots \\
&*(1 - \text{sncdf}( (\text{grid(ngrid)} - \text{means})/\text{sds} )) \ldots \\
\end{align*}\]

\% DETERMINE NEW THRESHOLD

\[\begin{align*}
\text{vwptr, tptr} &= \text{max}( \text{min}( 0, \text{Vjp1-Wjp1} ) ) ; \\
&\quad \% \text{Nearest right grid point to Vjp1-WJp1=0.} \\
\text{tptr} &= \text{tptr-1}; \quad \% \text{Nearest point on the left} \\
\text{thresp1} &= \text{Wjp1*( grid(tptr+1)-grid(tptr) )/( Vjp1(tptr+1)-Vjp1(tptr) ) } \ldots \\
&\quad \% \text{thresp1} = \text{grid(tptr+1)*Vjp1(tptr+1)} \ldots \\
&\quad \% \text{thresp1} = \text{grid(tptr+1)*Vjp1(tptr+1)} \ldots \\
&\quad \% \text{thresp1} = \text{grid(tptr+1)*Vjp1(tptr+1)} \ldots \\
&\quad \% \text{thresp1} = \text{grid(tptr+1)*Vjp1(tptr+1)} \ldots \\
&\text{dist} = 100*2*\text{abs( thresp1-thres + min(rlspan,2)- 2 )/abs(thresp1+thres);} \\
\end{align*}\]

\% UPDATE VALUE FUNCTIONS AND THRESHOLD VALUE

\[\begin{align*}
\text{if rlspan} &= \text{rlspan+1}; \\
\text{Vj} &= (\text{Vjp1*0.4+0.6*Vj}); \quad \% \text{Update married} \\
\text{Wj} &= (\text{Wjp1*0.4+0.6*Wj}); \quad \% \text{Update single} \\
\text{else} \\
\text{Vj} &= \text{Vjp1}; \quad \% \text{Update married} \\
\text{Wj} &= \text{Wjp1}; \quad \% \text{Update single} \\
\text{end} \\
\text{thres} &= \text{thresp1}; \quad \% \text{Update threshold} \\
\text{rlspan} &= \text{rlspan+1}; \quad \% \text{Update lifespan} \\
\end{align*}\]

\%___________________________STATIONARY DISTRIBUTION____________________________

\% Take a finer grid over the same end points

\[\begin{align*}
\text{ngridp} &= 5000; \quad \% \text{Number of grid points} \\
\text{gridp} &= [\text{mqlow: (span)/(ngridp-1): mqhigh}]; \quad \% \text{Construct grid} \\
\text{[vptr, nptr]} &= \text{max}( \text{min}(0,\text{gridp-thres} ) ); \\
&\quad \% \text{nptr is the first grid point to the right of thres}
\end{align*}\]
% DISTRIBUTION FOR SINGLES

probs = zeros(ngridp,1);

h = (gridp(2) - gridp(1))/2;  % Distance term
probs(1) = sncdf ((gridp(1) + h - means)/sds);
probs(ngridp)=1-sncdf( (gridp(ngridp) - h - means)/sds);
probs(2:ngridp-1) = sncdf( (gridp(2:ngridp-1) + h - means)/sds) ...
    - sncdf((gridp(2:ngridp-1)- h-means)/sds);

% TRANSITION MATRIX FOR MARRIED AGENTS

probm = zeros(ngridp,ngridp);       % The transition matrix: state i into j

for j=1:ngridp;

    probm(j,1) = sncdf ((gridp(1) - rho*gridp(j) + h - meanm)/sdm);
    probm(j,ngridp) = 1 - sncdf((gridp(ngridp)-rho*gridp(j)-h-meanm)/sdm);
    probm(j,2:ngridp-1)=(sncdf((gridp(2:ngridp-1)-rho*gridp(j)+h-
        meanm)/sdm)...
        - sncdf((gridp(2:ngridp-1)- rho*gridp(j)- h-meanm)/sdm )
    )';

end

% STATIONARY DISTRIBUTION FOR MARRIED AGENTS

% construct initial guess

mugridx = gridp(nptr:ngridp);
    mugrid = gridp;
    [nmugrid,n2]=size(mugrid);
    [nmugridx,n2]=size(mugridx);

muprobin=[ones(nmugridx,1)*(1/nmugridx)*0.5]; % Initial guess
    muprobin=[zeros(nptr-1,1);muprobin];

d = 1;                                    % Distance measure

while d > 0.00001;

    m = sum(muprobin); % Number of married agents

    muprobup = ( (1-m) + delta*m )*probs(nptr:ngridp) + ... (1-
        delta)*(probm(nptr:ngridp,nptr:ngridp)'*muprobin(nptr:ngridp));

    muprobup=[zeros(nptr-1,1);muprobup];

% Iterate on equation defining stationary distribution

d = norm(muprobup-muprobin)/norm(muprobin);

muprobin = muprobin;  % Update

end

muprob = muprobup;

pnmar = sum(probs(nptr:ngridp));  % prob of a new marriage
prmar = sum(probm(nptr:ngridp,nptr:ngridp)’*muprobin(nptr:ngridp))  
   / (sum(muprobup));  % prob of remaining married
% trans.m
% __________________________________________ TRANSITIONAL DYNAMICS _______________________

year   = sdate:1:edate; % Vector of years

umv = umvec;
usv = usvec;

threst = zeros(1,edate-sdate+1);  % Vec to hold thresholds
marrt  = zeros(1,edate-sdate+1);  % Vec to hold fraction of marrs
probdt = zeros(1,edate-sdate+1);  % Vec to hold prob of div for marrs
probmt = zeros(1,edate-sdate+1);  % Vec to hold prob of marr for sing
durmt  = zeros(1,edate-sdate+1);  % Vec to dur of marr.

Vj   = Vjn;    % Married value function from FINAL steady state
Wj   = Wjn;    % Single value function from FINAL steady state
thres   = thresn; % Threshold from FINAL steady state

% COMPUTE TIME SERIES FOR THRESHOLDS STARTING BACKWARDS FROM THE END DATE

disp('Computing Thresholds Working Backwards in Time')

for j=1:edate-sdate+1;                   % Begin iterating from end date
    %disp( year(edate - sdate - j + 1) ) % Display year
    [vptr, nptr] = max( min(0,grid-thres) );
        % Nearest rght grid pnt to "thres"
    nptr = nptr-1;
        % Nearest left grid pnt to "thres"
    nptr = max(nptr,1);
    lower = [thres, grid(nptr+1:ngrid-1)']';
        % Column vector of lower bounds
    upper = grid(nptr+1:ngrid);
        % Column vector of upper bounds
    condms = nmean(lower,upper,0,means,vars);
        % Row vec of cond means -- single

    diffvj = Vj(nptr+1:ngrid) - Vj(nptr:ngrid-1);    % First diffs of Vj
    diffg  = grid(nptr+1:ngrid) - grid(nptr:ngrid-1);
        % First diffs of grid
    dervj  = diffvj./diffg;
        % Slope of v func for marrieds
    intcpt = ( Vj(nptr:ngrid-1).*grid(nptr+1:ngrid) ... 
               - Vj(nptr+1:ngrid).*grid(nptr:ngrid-1) )./diffg;
    dcdffs = cdfms(nptr+1:ngrid) ... 
             - [sncdf( (thres-means)/sds ), cdfms(nptr+1:ngrid-1)];  % First diff of single CDF
    rtailsm = nmean(grid(ngrid),1000,0,means,vars);
for i = 1:ngrid;

cdfmm = sncdf((grid-rho*grid(i)-meanm)/sdm)';
% Cdf on grid for married agents
dcddfm = cdfmm(nptr+1:ngrid)- ... 
[sncdf((thres-rho*grid(i)-meanm)/sdm), cdfmm(nptr+1:ngrid-1)];
% First difference of married CDF

condmm = nmean(lower,upper,grid(i),meanm,varm);
% Row vector of cond means -- married
rtaillmm = nmean(grid(ngrid),1000,grid(i),meanm,varm);
% Cond mean for rght tail -- married

Vjp1(i) = umv(edate-sdate-j+2)+grid(i)... 
+ beta*sncdf((thres-rho*grid(i)-meanm)/sdm)*Wj ...
+ beta*dcdfm*intcpt + beta*condmm*dervj ...
+ beta*( ( Vj(ngrid)-Vj(ngrid-1) )/(grid(ngrid)-grid(ngrid-1))*rtaillmm ...
+((grid(ngrid)*Vj(ngrid-1)-grid(ngrid-1)*Vj(ngrid))... 
/(grid(ngrid)-grid(ngrid-1)))*rtaillmm ...
*(1-sncdf( (grid(ngrid) - rho*grid(i)-meanm)/sdm) ) ) ;

end

Wjp1 = usv(edate-sdate-j+2)+beta*sncdf((thres-means)/sds)*Wj ...
+ beta*dcdfs*intcpt + beta*condms*dervj ...
+ beta*( ( Vj(ngrid)-Vj(ngrid-1) )/(grid(ngrid)-grid(ngrid-1))*rtailsm ...
+((grid(ngrid)*Vj(ngrid-1)-grid(ngrid-1)*Vj(ngrid))... 
/(grid(ngrid)-grid(ngrid-1)))*rtailsm ...
*(1-sncdf( (grid(ngrid) - means)/sds) ) ;

% DETERMINE NEW THRESHOLD

[vwptr, tptr] = max( min( 0, Vjp1-Wjp1 ) );
% Nearest right grid point to Vjp1-WJp1=0.
tptr = tptr-1;
% Nearest point on the left
tptr = max(tptr,1);
theresp1 = Wjp1*( grid(tptr+1)-grid(tptr))/( Vjp1(tptr+1)-Vjp1(tptr) ) ...
+ ( grid(tptr)*Vjp1(tptr+1) - grid(tptr+1)*Vjp1(tptr) ) ... 
/( Vjp1(tptr+1) - Vjp1(tptr) ) ;
dist = 100*norm( log(Vjp1)-log(Vj) );
% Comp dist between values--in %

% UPDATE VALUE FUNCTIONS AND THRESHOLD VALUE
Vj = Vjp1; % Update married
Wj = Wjp1; % Update single
threst(edate - sdate - j + 2) = thresp1; % Save thresholds
thres = thresp1; % Update threshold

end

plot(year, threst)
title('The Behavior of the Threshold over Time')
xlabel('Year')
ylabel('Threshold')

% COMPUTE THE TIME SERIES FOR THE MARRIED
% PROB DIST MOVING FORWARD FROM THE START DATE

muprobin = muprobo; % Initial guess

disp('Computing Distribution for Married Agents Moving Forward in Time')

for j = 1:edate-sdate+1; % Begin iterating from start date

    [vptr, nptr] = max( min(0, gridp-threst(j)) );
    % First grid point on the right of threshold

    m = sum(muprobin);
    % Number of married agents -- before current shock is drawn

    muprobup = ( (1-m) + delta*m)*probs(nptr:ngridp) ... 
    + (1-
    delta)*probm(nptr:ngridp,nptr:ngridp)'*muprobin(nptr:ngridp);
    % Probability dist over m. agents -- after shock is drawn

    muprobup = [zeros(nptr-1,1);muprobup];
    % Iterate on equation defining stationary distribution
d = norm(muprobup-muprobin)/norm(muprobin);

    muprobin = muprobup; % Update

    mp = sum(muprobin);

    marrt(j) = sum(muprobin); % Record number of married agents -- beginning of period

    probdt(j)=1-
    sum(prob(nptr:ngridp,nptr:ngridp)'*muprobin(nptr:ngridp))/mp;
    % Divorce Rate

    probmt(j) = sum( probs(nptr:ngridp) );
% Marriage Rate
% Number of married agents -- end of period

probd=1 - sum(probm(nptr:ngridp,nptr:ngridp)'*muprobin(nptr:ngridp))/mp;

probremainsingle = 1 - probmt(j);
probremainmar = 1 - probd;

durmt(j)  = 1/(1-probremainmar*(1-delta));

% Duration of marriage -- assuming current divorce rate

durst(j)=1/ (1-probremainsingle*(1-delta)); % Durat of sing state

fracmt(j)=durmt(j)/(durmt(j)+durst(j));  % Frac of Life Spent mar
end

load maritalstatusdata2.txt;  % Martial status
load mardivrates.txt;         % Divorce rates

disp('================================')
disp('================================')
plot(mugrid, muprobup, mugrid, muprobn)
title('  Distribution over Marr. Agents--terminal year vs final s.
state')
xlabel('Year')
ylabel('Probability')
pause(5)

plot(year, probdt, year, probmt)
title('  Probabilities of Divorce and Marriage')
xlabel('Year')
ylabel('Probability')
pause(5)

plot(year, durmt)
title('  Duration of Marr. -- assuming current div. rate')
xlabel('Year')
ylabel('Duration')
pause(5)

plot(year, durst)
title('  Duration of Single State -- assuming current div. rate')
xlabel('Year')
ylabel('Duration')
pause(5)
plot(year, fracmt)
title(' Fraction of Life Spent Married')
xlabel('Year')
ylabel('Fraction')
pause(5)

plot(year, marrt)
title(' Fraction of Agents who are Married')
xlabel('Year')
ylabel('Fraction')
hold
plot(year, maritalstatusdata2,'r')
disp('================================')
disp('================================')
disp('FIGURE 6 in the paper')
disp('PRESS ANY KEY')
pause
hold off

plot(year, probdt)
title(' Divorce Rate')
xlabel('Year')
ylabel('Probability')
hold
plot(years(1:47),mardivrates(:,2),'r')
disp('================================')
disp('================================')
disp('FIGURE 7 in the paper')
disp('PRESS ANY KEY')
pause
hold off

plot(year, probmt)
title(' Marriage Rate')
xlabel('Year')
ylabel('Probability')
hold
plot(years(1:47),mardivrates(:,3),'r')
disp('FIGURE 7 in the paper')
disp('PRESS ANY KEY')
pause
hold off
<table>
<thead>
<tr>
<th>Year</th>
<th>mardivrates</th>
<th>0.211202306</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950</td>
<td>0.010993718</td>
<td>0.211202306</td>
</tr>
<tr>
<td>1951</td>
<td>0.010689636</td>
<td>0.210548455</td>
</tr>
<tr>
<td>1952</td>
<td>0.010878011</td>
<td>0.198570433</td>
</tr>
<tr>
<td>1953</td>
<td>0.010674403</td>
<td>0.208693305</td>
</tr>
<tr>
<td>1954</td>
<td>0.010256271</td>
<td>0.198006645</td>
</tr>
<tr>
<td>1955</td>
<td>0.010079944</td>
<td>0.207649532</td>
</tr>
<tr>
<td>1956</td>
<td>0.010168228</td>
<td>0.208854922</td>
</tr>
<tr>
<td>1957</td>
<td>0.010089332</td>
<td>0.202319072</td>
</tr>
<tr>
<td>1958</td>
<td>0.009609861</td>
<td>0.189524556</td>
</tr>
<tr>
<td>1959</td>
<td>0.0101854</td>
<td>0.196967699</td>
</tr>
<tr>
<td>1960</td>
<td>0.010030585</td>
<td>0.193004689</td>
</tr>
<tr>
<td>1961</td>
<td>0.010483667</td>
<td>0.188206687</td>
</tr>
<tr>
<td>1962</td>
<td>0.010334559</td>
<td>0.192363991</td>
</tr>
<tr>
<td>1963</td>
<td>0.010568423</td>
<td>0.196297175</td>
</tr>
<tr>
<td>1964</td>
<td>0.011009174</td>
<td>0.201119272</td>
</tr>
<tr>
<td>1965</td>
<td>0.011599748</td>
<td>0.199933356</td>
</tr>
<tr>
<td>1966</td>
<td>0.011958111</td>
<td>0.19657034</td>
</tr>
<tr>
<td>1967</td>
<td>0.012410422</td>
<td>0.196893839</td>
</tr>
<tr>
<td>1968</td>
<td>0.013737298</td>
<td>0.200678952</td>
</tr>
<tr>
<td>1969</td>
<td>0.014818079</td>
<td>0.202167766</td>
</tr>
<tr>
<td>1970</td>
<td>0.016244866</td>
<td>0.19450419</td>
</tr>
<tr>
<td>1971</td>
<td>0.017601785</td>
<td>0.187653645</td>
</tr>
<tr>
<td>1972</td>
<td>0.018927091</td>
<td>0.192521849</td>
</tr>
<tr>
<td>1973</td>
<td>0.020278356</td>
<td>0.185112894</td>
</tr>
<tr>
<td>1974</td>
<td>0.021535477</td>
<td>0.170138649</td>
</tr>
<tr>
<td>1975</td>
<td>0.022816368</td>
<td>0.154345881</td>
</tr>
<tr>
<td>1976</td>
<td>0.023602999</td>
<td>0.145536066</td>
</tr>
<tr>
<td>1977</td>
<td>0.023782535</td>
<td>0.137679623</td>
</tr>
<tr>
<td>1978</td>
<td>0.0246892</td>
<td>0.134520335</td>
</tr>
<tr>
<td>1979</td>
<td>0.025706885</td>
<td>0.132281945</td>
</tr>
<tr>
<td>1980</td>
<td>0.025724238</td>
<td>0.129091164</td>
</tr>
<tr>
<td>1981</td>
<td>0.025588018</td>
<td>0.121826024</td>
</tr>
<tr>
<td>1982</td>
<td>0.024563834</td>
<td>0.118346326</td>
</tr>
<tr>
<td>1983</td>
<td>0.024267063</td>
<td>0.113548333</td>
</tr>
<tr>
<td>1984</td>
<td>0.024407558</td>
<td>0.111304457</td>
</tr>
<tr>
<td>1985</td>
<td>0.024767415</td>
<td>0.106254955</td>
</tr>
<tr>
<td>1986</td>
<td>0.024276146</td>
<td>0.103906544</td>
</tr>
<tr>
<td>1987</td>
<td>0.023832884</td>
<td>0.102154036</td>
</tr>
<tr>
<td>1988</td>
<td>0.023717584</td>
<td>0.100243755</td>
</tr>
<tr>
<td>1989</td>
<td>0.023552163</td>
<td>0.097365312</td>
</tr>
<tr>
<td>1990</td>
<td>0.023838816</td>
<td>0.09755226</td>
</tr>
<tr>
<td>1991</td>
<td>0.023938691</td>
<td>0.091622227</td>
</tr>
<tr>
<td>1992</td>
<td>0.024402</td>
<td>0.089826963</td>
</tr>
<tr>
<td>1993</td>
<td>0.023694507</td>
<td>0.087507499</td>
</tr>
<tr>
<td>1994</td>
<td>0.023633766</td>
<td>0.085221533</td>
</tr>
<tr>
<td>1995</td>
<td>0.022848096</td>
<td>0.084271284</td>
</tr>
<tr>
<td>1996</td>
<td>0.022538855</td>
<td>0.082121711</td>
</tr>
</tbody>
</table>
maritalstatusdata2.txt

0.8161  
0.8247  
0.8230  
0.8314  
0.8308  
0.8353  
0.8319  
0.8353  
0.8334  
0.8364  
0.8323  
0.8276  
0.8298  
0.8278  
0.8266  
0.8210  
0.8154  
0.8115  
0.8048  
0.8025  
0.7970  
0.7900  
0.7902  
0.7853  
0.7759  
0.7650  
0.7560  
0.7435  
0.7296  
0.7227  
0.7140  
0.7045  
0.6965  
0.6890  
0.6827  
0.6791  
0.6769  
0.6753  
0.6731  
0.6656  
0.6644  
0.6571  
0.6544  
0.6526  
0.6452  
0.6486  
0.6413  
0.6338  
0.6314  
0.6250  
0.6249
<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.00</td>
</tr>
<tr>
<td>101.14</td>
</tr>
<tr>
<td>102.65</td>
</tr>
<tr>
<td>105.80</td>
</tr>
<tr>
<td>105.25</td>
</tr>
<tr>
<td>110.33</td>
</tr>
<tr>
<td>113.57</td>
</tr>
<tr>
<td>114.31</td>
</tr>
<tr>
<td>113.47</td>
</tr>
<tr>
<td>116.58</td>
</tr>
<tr>
<td>117.13</td>
</tr>
<tr>
<td>119.05</td>
</tr>
<tr>
<td>122.93</td>
</tr>
<tr>
<td>125.73</td>
</tr>
<tr>
<td>132.98</td>
</tr>
<tr>
<td>139.45</td>
</tr>
<tr>
<td>145.20</td>
</tr>
<tr>
<td>149.76</td>
</tr>
<tr>
<td>155.11</td>
</tr>
<tr>
<td>158.35</td>
</tr>
<tr>
<td>163.08</td>
</tr>
<tr>
<td>168.10</td>
</tr>
<tr>
<td>174.13</td>
</tr>
<tr>
<td>184.36</td>
</tr>
<tr>
<td>181.46</td>
</tr>
<tr>
<td>183.05</td>
</tr>
<tr>
<td>189.15</td>
</tr>
<tr>
<td>194.02</td>
</tr>
<tr>
<td>201.51</td>
</tr>
<tr>
<td>205.01</td>
</tr>
<tr>
<td>204.29</td>
</tr>
<tr>
<td>206.85</td>
</tr>
<tr>
<td>207.66</td>
</tr>
<tr>
<td>212.04</td>
</tr>
<tr>
<td>226.36</td>
</tr>
<tr>
<td>231.83</td>
</tr>
<tr>
<td>237.07</td>
</tr>
<tr>
<td>239.98</td>
</tr>
<tr>
<td>248.28</td>
</tr>
<tr>
<td>252.23</td>
</tr>
<tr>
<td>255.09</td>
</tr>
<tr>
<td>252.74</td>
</tr>
<tr>
<td>257.15</td>
</tr>
<tr>
<td>257.34</td>
</tr>
<tr>
<td>260.80</td>
</tr>
<tr>
<td>264.50</td>
</tr>
<tr>
<td>267.99</td>
</tr>
<tr>
<td>273.05</td>
</tr>
<tr>
<td>284.36</td>
</tr>
<tr>
<td>288.41</td>
</tr>
<tr>
<td>298.88</td>
</tr>
</tbody>
</table>