```
new;
output file=LTC_AER_program1.out reset;
print "start time";
time;

/*-----------------------------------------------------*/
/* This GAUSS program accompanies Brown & Finkelstein  */
/* "The Interaction of Public and Private Insurance"   */
/* Medicaid and the Long-Term Care Insurance Market"   */
/* in the American Economic Review                     */
/*                                                     */
/* The program calculates the LTCI values found in     */
/* Figures 1 and 2, as well as data for Table 2        */
/*-----------------------------------------------------*/

format /rdt 1,5;
/***************************************************************************/
/*      VARIABLES TO BE INPUTED BY USER                                    */
/***************************************************************************/

bencap=1;       @ =1 if benefit cap applies. =0 if policy benefits are uncapped @;
do while bencap<=1;

X=1000;         @ Wealth scaling factor - converts to units of $X              @

male=1;         @ =1 if male, =0 if female          @
do while male>=0;

beta=0;         @ Fraction of HC expenses that enter into utility function as consumption   @;

qual2=1;        @ Medicaid quality for state 2: Typically equal to 1 unless there is stigma effect @;
qual3=1;        @ Medicaid quality for state 3 @;
qual4=1;        @ Medicaid quality for state 4 @;

                @ Next parameters =1 unless state dependent utility is desired @;
phi1=1;         @ coefficient on utility of consumption in state 1  @;
phi2=1;         @ coefficient on utility of consumption in state 2  @;
phi3=1;         @ coefficient on utility of consumption in state 3  @;
phi4=1;         @ coefficient on utility of consumption in state 4  @;

beq=0;          @ coefficient on utility of bequests @;
fo=0;           @ if fo=1, policy is facility only, meaning it covers only states 3 and 4.@;
do while fo<=0; @ if fo=0, policy covers HC (state 2) also @

wcount=3;               @ This tells program which wealth deciles to loop through.              @;
do while wcount<=9;     @ for each case, starting wealth is defined by "wealth"                 @;
                        @ alpha is the fraction of total wealth annuitized                      @;
if wcount==0;           @ wx is used purely to adjust upper bound of grid for discretization    @;
    wealth=40000;       @ if LTCI value gets high (e.g., higher risk aversion), adjust wx up    @;
    alpha=.98;          @ grid = grid size used for discretization.                             @;
    wx=0;
    grid=20;
elseif wcount==1;
    wealth=58450;
    alpha=.98;
    wx=0;
    grid=20;
elseif wcount==2;
    wealth=93415;
    alpha=.91;
    wx=20000;
    grid=40;
elseif wcount==3;
    wealth=126875;
    alpha=.82;
    wx=30000;
    grid=75;
elseif wcount==4;
    wealth=169905;
    alpha=.70;
    wx=10000;
    grid=100;
elseif wcount==5;
    wealth=222570;
    alpha=.60;
    wx=50000;
    grid=130;
elseif wcount==6;
    wealth=292780;
    alpha=.52;
    wx=20000;
    grid=175;
elseif wcount==7;
    wealth=385460;
    alpha=.41;
    wx=40000;
    grid=225;
elseif wcount==8;
    wealth=525955;
    alpha=.35;
    wx=40000;
    grid=300;
elseif wcount==9;
    wealth=789475;
    alpha=.26;
    wx=75000;
    grid=450;
endif;

w0=wealth*(1-alpha);                    @ This section simply creates discretized grid @;
ww=round((w0*1.2-10000)/grid);
if ww*grid<10000;
    wdis=seqa(0,10,100)|seqa(1000,20,50)|seqa(2000,50,160);
else;
    wdis=seqa(0,10,100)|seqa(1000,20,50)|seqa(2000,50,160)|seqa(10000,grid,ww);
endif;
wdis=wdis/X;
w0=w0/X;
wrow=rows(wdis);
```

```
neginf=-999999999999;       @ Felicity value of zero consumption (negative infinity)        @
age=65;                     @ Age at time of purchase               @
maxage=105;                 @ Max age for calculation           @

tn=(maxage-age)*12;         @ number of periods in the problem  @

graphs = 0;                 @put in the # of different health trajectories you want to graph@

gam=3;                      @ coefficient of relative risk aversion     @
do while gam<=3.1;          @ Note that when gam=1, undefined.  So uses value "approaching" 1 @;
gam=trunc(gam);

Food=515/X;                 @ Food = SSI level used to parameterize food/housing benefit @
do while Food<=515/X;

Mcaid=1;                    @ =1 if there is a Medicaid program     @

medscale=1;                 @ scaling variable used for sensitivity checks.  Typically set = 1 @;
do while medscale<=1;

Wbar=2000/X;                @ wealth excluded from Medicaid spend-down (base case is 2000/X)    @
Cbar=30/X;                  @ min consumption provided if on Medicaid   @
Cbar2=545/X;                @ CBAR WHILE IN HOME CARE @;

Wbar=medscale*Wbar;
/*
Cbar=medscale*Cbar;
Cbar2=medscale*Cbar2;
*/;

Medicare=.35;               @ Fraction of Home care costs covered by Medicare @
do while Medicare<=.35;


MWcount=0;                  @ Alternative scenarios for market loads of private LTCI prices @
do while MWcount<=0;

    if MWcount==0;          @ Base case market loads @
        if male==0;
            MW=1.058;
        elseif male==1;
            MW=0.5;
        else;
        endif;
    elseif MWcount==1;
        if male==0;
            MW=.6;
        elseif male==1;
            MW=0.3;
        else;
        endif;
    elseif MWcount==2;         @ actuarially fair on average but still with unisex pricing @
        if male==0;
            MW=1.358127;
        elseif male==1;
```

```
            MW=0.6418727;
        else;
        endif;
    else;                       @ Actuarially fair prices @
        MW=1;
    endif;


Binf=0;                     @ rate of nominal growth of benefits under LTCI policy  @;
Bben=3000/X;                @ Monthly benefit from LTCI policy                      @;
do while Bben<=(3000/X);


NHamt=4290/X;               @ Monthly cost of NH    @
ALFamt=2159/X;              @ Monthly cost of ALF   @

HCnonrn=18/X;               @ Hourly HC costs (non RN) @
HCrn=37/X;                  @ Hourly HC costs (RN)  @

r=.03;                      @ real interest rate                        @
rho=.03;                    @ discount rate for utility of consumption  @
d=.03;                      @ discount rate for utility of bequests     @
inf=.03;                    @ price inflation rate                      @
rminf=.015;                 @ real medical cost growth (over inflation) @

/***************************************************************************/
/*  END USER INPUT                                 */
/***************************************************************************/

if beq==1;          /* This speeds up procedure when no bequests */;
    wdisbeq=wdis;
else;
    wdisbeq=1;
endif;

Prem = 150/X;           @ Just a placeholder -- later we replace with fair premium @;

Minf=rminf+inf;         @ nominal medical price inflation  @

/* Convert annual growth rates into monthly growth rates */;

r=((1+r)^(1/12))-1;
d=((1+d)^(1/12))-1;
rho=((1+rho)^(1/12))-1;
inf=((1+inf)^(1/12))-1;
Minf=((1+Minf)^(1/12))-1;
Binf=((1+Binf)^(1/12))-1;

/*----------------------------------------------*/
/*      LOAD DATA MATRICES                      */
/*----------------------------------------------*/

if male==1;
    load q[]=c:/gaussrun/tranmi65.out;
                    @ matrix of transition probabilities    @
                    @ rows=ages 65 to 111                   @
                    @ col 1 is age.  col 2 through 26 are   @
                    @ transition probs from state i to j    @;

    load hcexp0[]=c:/gaussrun/HCexp0mi65.out;  @ load hc matrix @
    load hcexp1[]=c:/gaussrun/HCexp1mi65.out;
else;
    load q[]=c:/gaussrun/tranfi65.out;
    load hcexp0[]=c:/gaussrun/HCexp0fi65.out;
    load hcexp1[]=c:/gaussrun/HCexp1fi65.out;
endif;


/* Note: The next few lines add 30 rows of zeros to the transition and HC cost matrices.  The
reason is that the whole program was written assuming that these matrices started at age 35.  The
data files now have the matrices starting at 65.  So rows of zeros (which are later dropped anyway)
simply allowed existing program to continue functioning appropriately. */;

q=reshape(q,47,26);
q=zeros(30,26)|q;
hcexp0=reshape(hcexp0,47,2);
hcexp0=zeros(30,2)|hcexp0;
hcexp1=reshape(hcexp1,47,2);
hcexp1=zeros(30,2)|hcexp1;

/* Recall that the hcexp matrices have column 1=age, column 2 = expenses */;

hcexp0[.,2]=hcexp0[.,2].*(4.33333*HCnonrn);
hcexp1[.,2]=hcexp1[.,2].*(4.33333*HCrn);
hcexp=hcexp0[.,1]~(hcexp0[.,2]+hcexp1[.,2]);
hcexp=(1-Medicare)*hcexp;
hcexp=hcexp[(age-35)+1:(maxage-35)+1,2].*.ones(12,1);
hcexp=hcexp[1:tn,1];


/*  HCutil is the amount of non-skilled NH care received, regardless of payer */;
HC0=hcexp0[(age-35)+1:(maxage-35)+1,2].*.ones(12,1)+hcexp1[(age-35)+1:(maxage-35)+1,2].*.ones(12,1);
HC0=HC0[1:tn,1];

@ B = Benefit matrix                @
@ P = Premium matrix                @
@ M = medical expenditure matrix    @
@ Rows are periods, cols are states @

M=zeros(tn,1)~hcexp~ALFamt*ones(tn,1)~NHamt*ones(tn,1)~zeros(tn,1);

if fo==0;  @ not facility only, so benefits paid in all sick states @;
    B=zeros(tn,1)~Bben*ones(tn,3)~zeros(tn,1);
        P=Prem*(ones(tn,1)~zeros(tn,4));
elseif fo==1;  @ if facility only, so benefits paid only in alf or nh @;
    B=zeros(tn,2)~Bben*ones(tn,2)~zeros(tn,1);
        P=Prem*(ones(tn,2))~zeros(tn,3);
else;
endif;
```

```
@ Incorporate nominal growth rates @
bfactor=cumprodc((1+Binf)*ones(tn,1));
mfactor=cumprodc((1+Minf)*ones(tn,1));
B=B.*bfactor;
M=M.*mfactor;
HC0=HC0.*mfactor;

@ Now take the min of benefit level and actual cost @;
mcomp=dotfle(M,B);  @mcomp=(M.<=B)@
bcomp=dotflt(B,M);  @bcomp=(B.<M)@

if bencap==1;
    if sumc(vecr(mcomp+bcomp))==(rows(B)*cols(B));
        B=(M.*mcomp)+(B.*bcomp);
    else;
        print "mistake";
    endif;
elseif bencap==0;
    B=M;
else;
    print "need to define bencap as 0 or 1";
endif;

@ Now convert from nominal to real @;
ifactor=cumprodc((1+inf)*ones(tn,1));

P=P./ifactor;
B=B./ifactor;
M=M./ifactor;
HC0=(HC0)./ifactor;

/*----------------------------------------------*/
/*  CREATE MONTHLY TRANSITION MATRIX    */
/*----------------------------------------------*/

q=q[(age-35)+1:(maxage-35)+1,2:26].*.ones(12,1);

/*----------------------------------------------*/
/*  CREATE ACTUARIALLY FAIR PREMIUM         */
/*----------------------------------------------*/

/*Turn conditional probabilities into unconditional probabilies*/
prob = zeros(rows(q) + 1,5);            @initialize@
prob[1,1] = 1;              @everyone starts out of care @

for i(2,rows(q),1);
      prob[i,1] = q[i-1,1]*prob[i-1,1] + q[i-1,6]*prob[i-1,2] + q[i-1,11]*prob[i-1,3] + q[i-1,16]*prob[i-1,4];
      prob[i,2] = q[i-1,2]*prob[i-1,1] + q[i-1,7]*prob[i-1,2] + q[i-1,12]*prob[i-1,3] + q[i-1,17]*prob[i-1,4];
      prob[i,3] = q[i-1,3]*prob[i-1,1] + q[i-1,8]*prob[i-1,2] + q[i-1,13]*prob[i-1,3] + q[i-1,18]*prob[i-1,4];
      prob[i,4] = q[i-1,4]*prob[i-1,1] + q[i-1,9]*prob[i-1,2] + q[i-1,14]*prob[i-1,3] + q[i-1,19]*prob[i-1,4];
      prob[i,5] = q[i-1,5]*prob[i-1,1] + q[i-1,10]*prob[i-1,2] + q[i-1,15]*prob[i-1,3] + q[i-1,20]*prob[i-1,4] + prob[i
endfor;


rfactor=cumprodc((1+r).*ones(rows(q)-12,1));
```

```
cost = P[.,.].*prob[2:rows(prob)-12,.];
cost = cost./rfactor;           @Discount@
ben = B[.,.] .*prob[2:rows(prob)-12,.];

ben = ben./rfactor;         @Discount@

scost = sumc(sumc(cost));
sben = sumc(sumc(ben));

AFP = (sben/scost)*prem;        @actuarially fair monthly premium@
actfprem=AFP[1,1];

/* Now replace P with actuarially fair premium * Money's Worth */

P=(P/Prem)*(AFP/MW);

clear cost;
clear scost;
clear ben;

/* Calculate monthly annuity amount     */

A=((alpha/(1-alpha))*w0)/sumc((1-prob[2:rows(prob)-12,5])./rfactor);
lifexp=sumc(1-prob[2:rows(prob)-12,5])/12;

clear prob;

/* Create LTCIown, a matrix that is tn rows x 5 states that is the net payment    */
/* to the policy holder.  Will be equal to -P when healthy, equal to benefit when */
/* receiving benefit and premium is waived, and equal to B-P when receiving       */
/* benefit but policy is not waived.                                              */
/* LTCInone is a matrix of zeros of same size as LTCIown to be used when the      */
/* individual does not own insurance                                              */

LTCIown=B-P;
LTCInone=zeros(rows(LTCIown),cols(LTCIown));

/*----------------------------------------------*/
/*      UTILITY FUNCTIONS                       */
/*----------------------------------------------*/

if gam==1;
        gam=1.00000001;                 /* If gamma=1, util is "undefined" */
else;                                   /* though it really equals natural */
        gam=gam;                        /* log in the limit                */
endif;

fn util(c)=(((c)^(1-gam))-1)/(1-gam);  /* CRRA Utility for health state=1    */
fn utilbeq(c)=beq*c;

/* fn utilbeq(c)=beq*(((c+.00000000001)^(1-gam))-1)/(1-gam); CRRA Utility for bequests if state=5    */;

/* So now we calculate maximum lifetime expected utility given that person owns an LTCI policy */;
```

```
{Vstar, Mstar, EPDVMedical, Istarown}=utilopt(LTCIown,1);   @ Note that utilopt procedure is defined below @;

/*
=================================================================================
       FINDING WEALTH EQUIVALENT
=================================================================================
  This section asks the question:  "Purchasing a LTCI policy gives me utility
       of Vstar.  Now suppose I take away the LTCI.  What level of wealth must
       I have to attain the level of utility Vstar without access to LTCI?"     */

clear Vmax;
clear V;
clear VV;
clear conrow;

clear Medicaid;
clear Medicaid2;


/* Now have procedure run through, calculating the Utility you get from
having NO LTCI policy, and have it report the entire vector.  In other words,
have it report the max utility achievable from following optimal consumption
path given every level of starting wealth from 0 to wmax */;


w0=wealth*(1-alpha);
ww=round((w0*1.2+wx-10000)/grid);
if ww*grid<10000;
    wdis=seqa(0,10,100)|seqa(1000,20,50)|seqa(2000,50,160);
else;
    wdis=seqa(0,10,100)|seqa(1000,20,50)|seqa(2000,50,160)|seqa(10000,grid,ww);
endif;
wdis=wdis/X;
w0=w0/X;
wrow=rows(wdis);

if beq==1;
    wdisbeq=wdis;
else;
    wdisbeq=1;
endif;

{Ustar, MUstar, EPDVMed2, Istarnone} =utilopt(LTCInone,2);

/* Then we want to find the element in Ustar that is closest to Vstar,
and then interpolate with the nearest value, in order to hone in on the level
of wealth that would get you utility of Vstar */;

Vr=sumc(dotfgt(Vstar,Ustar));

if Vr==rows(UStar);
    print "Grid not big enough!";       @ if get this message, go back and adjust wx @;
    wequiv=neginf;
elseif Vr==0;
    print "Vr==0:  LTCI is worth than losing all financial wealth.";
    wequiv=minf;
else;
    if Ustar[Vr+1]>=Vstar and Vstar>=UStar[Vr];
        wequiv=wdis[Vr]+((Vstar-Ustar[Vr])/(Ustar[Vr+1]-Ustar[Vr]))*(wdis[Vr+1]-wdis[Vr]);
        wequiv=(wequiv/(1+r))-w0;
    else;
        print "Ustar[1]~Ustar[Vr+1]~Vstar~UStar[Vr]~Ustar[rows(UStar)]";
        Ustar[1]~Ustar[Vr+1]~Vstar~UStar[Vr]~Ustar[rows(UStar)];
        wequiv=neginf;
    endif;
endif;

/* OUTPUT TO SCREEN */;

if male==0;
   print "gender =                                                   female";
else;
   print "gender =                                                    male";
endif;
print " ";
print "wealth decile                                               " wcount;
print " ";
print "Is there benefit cap? yes=1, 0=no                           " bencap;
print "Gross load                                                  " 1-MW;
print " ";


print "Medicaid share of EPDV of total LTC Exp (No private ins)   " MUstar/EPDVMedical;
print "     (Previous figure is column 1 of table 2)";
print " ";
print "Medicaid share of EPDV of total LTC Exp (With private ins) " Mstar/EPDVMedical;
print "     (Previous figure is column 2 of table 2)";
print " ";
print "implicit tax                                             " (MUstar - Mstar)/Istarown;
print "     (Previous figure is column 3 of table 4)";
print " ";
print "Net load                                                 " 1-(Istarown-(MUstar-Mstar))/(Istarown/MW);
print "     (Previous figure is column 4 of table 4)";
print " ";
print "Willingness to pay for private LTCI (figures 1 and 2)";
if rows(wequiv)==1;
    if Vr==rows(UStar);
        print "     Increase in wealth exceeds " wequiv;   @ if so, need to increase grid @;
    elseif Vr<=1;
        print "     LTCI is worse than losing all financial wealth " wequiv;
    else;
        print "     LTCI is equiv to increase in wealth of  " ((wdis[Vr]/(1+r))-w0)~wequiv*X~((wdis[Vr+1]/(1+r))-w0);
    endif;
else;
    print "     Too many rows of wequiv!";    @ signifies error  @;
endif;
print " ";
print " ";
print "ASSUMPTIONS AND INTERMEDIATE OUTPUT FOLLOWS ...";
print "Medicaid                                                 " Mstar;
print "Medicaid w/o LTCI                                        " MUstar;
print "EPDV of All Medical Costs                                " EPDVMedical;
print "EPDV of All Medical Costs w/o LTCI (should be same)      " EPDVMed2;
print "EPDV of Insurance benefits (should be same as sben)      " Istarown;
print "EPDV of Insurance benefits w/o insurance (should be 0)   " Istarnone;
print "Vstar                                                    " Vstar;
print "risk aversion is                                         " gam;
print "Total wealth and financial wealth (w0) are               " wealth~w0;
print "Fraction of total wealth annuitized and monthly annuity: " alpha~A;
print "Is it facility only policy yes=1, no=0                   " fo;
print "Fraction of HC covered by Medicare                       " Medicare;
print "age and maxage are                                       " age~maxage;
print "r, rho and delta are                                     " ((1+r)^12)-1~((1+rho)^12)-1~((1+d)^12)-1;
print "inflation is                                             " ((1+inf)^12)-1;
print "Medicaid? (=1 if yes) is                                 " Mcaid;
print "Food value                                               " Food;
print "Wbar, Cbar and Cbar2 are                                 " Wbar~Cbar~Cbar2;
print "Qual2, Qual3 and Qual4                                   " qual2~qual3~qual4;
print "A dollar in home care is worth Beta in consumption       " Beta;
print "LTCI benefit level is                                    " Bben;
print "Benefit inflation protection / growth is                 " ((1+Binf)^12)-1;
print "Starting medical expenditures for alf and nh are         " ALFamt~NHamt;
print "Bequest factor is                                        " beq;
print "State dependent utility (phi), states 1 - 4 private pay  " phi1~phi2~phi3~phi4;
print "Monthly premium when healthy is                          " P[1,1];
print "actuarially fair premium is                              " actfprem;
print "Actuarial value of benefits is                           " sben;
print "Money's worth                                            " MW;
print "life expectancy is                                       " lifexp;
print "SCaling factor X is                                      " X;
print "Discretization - # rows of wdis                          " rows(wdis);
print "If wequiv greater than this value, increase wx           " wx;
print "end time is";
time;

Bben=Bben+1500/X;
endo;
MWcount=MWcount+2;
endo;
Medicare=Medicare+.4;
endo;
medscale=medscale+5;
endo;
Food=Food+250/X;
endo;
gam=gam+2;
endo;
wcount=wcount+1;
endo;
fo=fo+1;
endo;
male=male-1;
endo;
bencap=bencap+1;
endo;
```

```
/*
=================================================================================
       PROCEDURE for COMPUTING UTILITY GIVEN LONG-TERM CARE MATRIX
=================================================================================

 NOTE:  This section defines a procedure called "utilopt" that is then called
        just like any other Gauss procedure.  It takes 2 arguments:
        LTCI is matrix of LTCI values.  If insurance exists, then
        call up using LTCIown as the LTCI matrix.  If no insurance, then use
        LTCInone (which is a matrix of zeros.  Proctype indicates if entire
        Vstar vector or just the optimal value should be the output of the procedure

A few other notes:
   1) Medicaid:  When one passes the asset and income test to qualify for Medicaid, the
   Medicaid vector keeps track of the amount of $ Medicaid is paying towards your care in that health
   and asset state.  This is updated every period to take into account the possibility of needing
   care tomorrow, and to do so uses the matrix Medicaid2.

2) Consumption and Asset paths:
   Matrices keep track of the potential consumption options in each period, and the program then
   picks the optimal consumption based on the vector of utility derived from each consumption level.
   Building on this, we define the matrix ASTREAM that keeps track of which "bin" in the descritized wealth
   vector is where that maximum is achieved.  Note: this matrix keeps track of the maximum for ANY HEALTH.WEALTH
   combination - so it has (tn*wrow) rows and 4 columns.  This info is used to define the CSTREAM matrix,
   which keeps track of the consumption at each of these maximum bins.  Again, this is a large matrix that can
   be used to trace out ANY consumption path for ANY health path.
   These matrices will be saved (such as for subsequent graphing) if the "graphs" variable >0.  */


proc(4)=utilopt(LTCI,proctype);
    local Vmax,B,c,cmaxind,cnegind,zeroind,VV;
    local t,wstart,wcompare,wcrit,Vstar,i,j,V,wextra;
    local Medicaid2, Medicaid, cell, Mstar;
    local insurance2, insurance, Istar;
    local Medical, Medicalstar;
    local Astream, Astream2, Cstream, Cstream1;
    local prec, field, fmat, s,w,g,l, amatrixname, cmatrixname;
    local gr, money;

    Vmax=zeros(wrow,4);                 /* Place holder for value function  */
    Medical = zeros(tn,5);


/* This section is simply a consistency check on the Medicaid programming*/
Medical[tn,.] = M[tn,1:5];
t = tn-1;
do while t >0;
    j = 1;
    do while j<=5;

        Medical[t,j] = M[t,j] + (1/(1+r))*(q[t+1,(j-1)*5+1]*Medical[t+1,1] + q[t+1,(j-1)*5+2]*Medical[t+1,2] + q[t+1,(j
        j = j+1;
    endo;
    t=t-1;
endo;

/*Now discount it back one more period*/
Medicalstar = (1/(1+r))*(q[1,1]*Medical[1,1] + q[1,2]*Medical[1,2] + q[1,3]*Medical[1,3] + q[1,4]*Medical[1,4]);

/*END OF CHECK*/

/* Create necessary matrices */;
    insurance = zeros(wrow,4);
    insurance2= zeros(wrow,4);
    Medicaid2 = zeros(wrow,4);
    Medicaid = zeros(wrow,4);

    Astream=zeros(tn*wrow,4);
    Cstream=zeros(tn*wrow,4);

    astream2=0;
    cstream1=0;

        /* Determines value function for period T */
        t=tn;
        i=1;                    /* i indexes wealth vector - amount of wealth at start of period */;
        do while i.<=wrow;
        j=1;                    /* j indexes health status 1 through 5 */
        do while j.<=4;

        VV=zeros(wrow,4);


/* Medicaid is defined so that (a) if Medicaid exists, and
(b) starting wealth for period falls below Wbar (asset test) and
(c) income plus LCTI income is less than sum of medical expenses and minimum
consumption level Cbar, then person permitted to consume max of Cbar. */;

/* Income test includes annuity, difference between insurance and expenses, +interest income)
Since wdis already includes the interest, the asset test must subtract interest back off.
The subtraction in the asset test and addition in the income test cancel in the combined constraint. */;

    if Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar2+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/(1+r))<(Cbar2) and j==2

        if wdis[i]/(1+r)>wbar;
            wextra=wdis[i]/(1+r)-wbar;
        else;
            wextra=0;
        endif;
        c=wdis[i]/(1+r)-wdis/(1+r)+Cbar2-wextra;

        Medicaid[i,j] = M[t,j] - wextra - (A+LTCI[t,j]+r*wdis[i]/(1+r) -cbar2);
        insurance[i,j] = LTCI[t,j];

        cnegind=(dotfgt(c,0));  @cnegind=(c.>0)@
        zeroind=1-cnegind;
        c=c.*(1-zeroind)+zeroind.*(.0001);
    elseif Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/(1+r))<(Cbar) and j>

        if wdis[i]/(1+r)>wbar;
            wextra=wdis[i]/(1+r)-wbar;
        else;
            wextra=0;
        endif;
            Medicaid[i,j] = M[t,j] - wextra - (A+LTCI[t,j]+r*wdis[i]/(1+r)-cbar);
            insurance[i,j] = LTCI[t,j];

        c=wdis[i]/(1+r)-wdis/(1+r)+Cbar-wextra;


        cnegind=(dotfgt(c,0));  @cnegind=(c.>0);@
        zeroind=1-cnegind;
        c=c.*(1-zeroind)+zeroind.*(.0001);

    else; @not on Medicaid@
        c=wdis[i]-M[t,j]+LTCI[t,j]+A-wdis/(1+r);
        Medicaid[i,j] = 0;
        if j >=2;
           insurance[i,j] = LTCI[t,j];
        endif;

        cmaxind=dotfge((wdis[i]+A-M[t,j]+LTCI[t,j]),c);     @.>=@
        cnegind=(dotfgt(c,0));      @cnegind=(c.>0);@
        zeroind=1-(cmaxind.*cnegind);
        c=c.*(1-zeroind)+zeroind.*(.0001);
    endif;


/* Note that the consumption vector will include negative values of consumption.  Thus there are
two constraints imposed -- one is that consumption be feasible, in that the individual's
starting wealth, annuity income, and net medical/LTCI payments must leave them with enough
income to consume the level c indicated.  The second is that consumption must be positive.
if either of these constraints fail, then the indicator zeroind replaces c with a small
positive value (.0001) just so that the CRRA utility function does not explode -- but
then later, it must replace the utility level associated with that consumption with neginf
(which stands for minus infinitiy - but is just a big negative number) so that we
are sure the program does not choose that path.  In some cases, these tests are redundant */;

/* Calculate value function for each combination of C(t) and W(t+1), given
that you came into this period with w(t) = wdis(i), and health status j */;

/* Note, the only reason it is necessary to do the following if-then statements is
if we one uses state dependent utility.  */;

            if j==1;
                    VV[.,j]=(phi1*util(c)+((1/(1+d))*utilbeq(wdisbeq))).*(1-zeroind)+zeroind*neginf;
            elseif j==2;
                if Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar2+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/(1+r))<(Cba
                    VV[.,j]=(phi2*util(c+qual2*beta*HC0[t,1])+((1/(1+d))*utilbeq(wdisbeq))).*(1-zeroind)+zeroind*neginf
                else;
                    VV[.,j]=(phi2*util(c+beta*HC0[t,1])+((1/(1+d))*utilbeq(wdisbeq))).*(1-zeroind)+zeroind*neginf;
                endif;
            elseif j==3;
                if Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/(1+r))<(Cbar
                    VV[.,j]=(phi3*util(c+qual3*Food)+((1/(1+d))*utilbeq(wdisbeq))).*(1-zeroind)+zeroind*neginf;
                else;
                    VV[.,j]=(phi3*util(c+Food)+((1/(1+d))*utilbeq(wdisbeq))).*(1-zeroind)+zeroind*neginf;
                endif;
            elseif j==4;
                if Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/(1+r))<(Cbar
                    VV[.,j]=(phi4*util(c+qual4*Food)+((1/(1+d))*utilbeq(wdisbeq))).*(1-zeroind)+zeroind*neginf;
                else;
                    VV[.,j]=(phi4*util(c+Food)+((1/(1+d))*utilbeq(wdisbeq))).*(1-zeroind)+zeroind*neginf;
                endif;
            else;
            endif;

/* For each possible combination of starting wealth for the period and health
status, choose the maximum value of VV */;

        Vmax[i,j]=maxc(VV[.,j]);

        Astream[((t-1)*wrow)+i,j] = indexcat(VV[.,j],Vmax[i,j]); @this picks the bin number where the maximum utility i

        Cstream[((t-1)*wrow)+i,j] = c[Astream[((t-1)*wrow)+i,j],1]; @this picks the consumption amt in the bin where th

        j=j+1;
        endo;
        i=i+1;
        endo;
        V=Vmax;
      Medicaid2 = Medicaid;
      insurance2= insurance;


/* V now represents the value function -- the max utility achievable from
following optimal consumption path in period tn.  Now
we go back to period tn-1, and ultimately back to time 0 */;

        t=tn-1;
        do while t>=1;
        Medicaid = zeros(wrow,4);
        insurance = zeros(wrow,4);


                i=1;                            /* i indexes t wealth state */
                do while i.<=wrow;
                    j=1;                        /* j indexes t health state */
                    do while j.<=4;

                        if Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar2+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/(1+

                            if wdis[i]/(1+r)>wbar;
                                wextra=wdis[i]/(1+r)-wbar;
                            else;
                                wextra=0;
                            endif;
            c=wdis[i]/(1+r)-wdis/(1+r)+Cbar2-wextra;

              Medicaid[i,j] = M[t,j] - wextra - (A+LTCI[t,j]+r*wdis[i]/(1+r)-cbar2);
            insurance[i,j] = LTCI[t,j];


            cnegind=(dotfgt(c,0));      @cnegind=(c.>0);@
            zeroind=1-cnegind;
            c=c.*(1-zeroind)+zeroind.*(.0001);
        elseif Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/

            if wdis[i]/(1+r)>wbar;
                wextra=wdis[i]/(1+r)-wbar;
            else;
                wextra=0;
            endif;
            c=wdis[i]/(1+r)-wdis/(1+r)+Cbar-wextra;

              Medicaid[i,j] = M[t,j] - wextra - (A+LTCI[t,j]+r*wdis[i]/(1+r) -cbar);
            insurance[i,j] = LTCI[t,j];

            cnegind=(dotfgt(c,0));      @cnegind=(c.>0);@
            zeroind=1-cnegind;
            c=c.*(1-zeroind)+zeroind.*(.0001);
 else; @not on Medicaid@
Medicaid[i,j] = 0;
if j>=2;
   insurance[i,j] = LTCI[t,j];
endif;
            c=wdis[i]-M[t,j]+LTCI[t,j]+A-wdis/(1+r);
            cmaxind=dotfge((wdis[i]+A-M[t,j]+LTCI[t,j]),c);     @.>=@
            cnegind=(dotfgt(c,0));      @cnegind=(c.>0);@
            zeroind=1-(cmaxind.*cnegind);
            c=c.*(1-zeroind)+zeroind.*(.0001);
        endif;

    if j==1;
        VV[.,j]=phi1*util(c).*(1-zeroind)+zeroind*neginf;
    elseif j==2;
        if Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar2+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/(1+
            VV[.,j]=(phi2*util(c+qual2*beta*HC0[t,1])).*(1-zeroind)+zeroind*neginf;
        else;
            VV[.,j]=(phi2*util(c+beta*HC0[t,1])).*(1-zeroind)+zeroind*neginf;
        endif;
    elseif j==3;
        if Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/(1+r
            VV[.,j]=phi3*util(c+qual3*Food).*(1-zeroind)+zeroind*neginf;
        else;
            VV[.,j]=phi3*util(c+Food).*(1-zeroind)+zeroind*neginf;
        endif;
    elseif j==4;
        if Mcaid==1 and (A+LTCI[t,j]+wdis[i]-M[t,j])<(Cbar+wbar) and (A+LTCI[t,j]-M[t,j]+r*wdis[i]/(1+r
            VV[.,j]=phi4*util(c+qual4*Food).*(1-zeroind)+zeroind*neginf;
        else;
            VV[.,j]=phi4*util(c+Food).*(1-zeroind)+zeroind*neginf;
```

```
                        endif;
                    else;
                    endif;

                /* Then, regardless of health status, we then have to add in the value of
                taking wealth into the next period, recognizing you can enter next period
                in any of 5 states.  */;

                VV[.,j]=VV[.,j]+(1/(1+rho))*q[t+1,(j-1)*5+1]*V[.,1];
                VV[.,j]=VV[.,j]+(1/(1+rho))*q[t+1,(j-1)*5+2]*V[.,2];
                VV[.,j]=VV[.,j]+(1/(1+rho))*q[t+1,(j-1)*5+3]*V[.,3];
                VV[.,j]=VV[.,j]+(1/(1+rho))*q[t+1,(j-1)*5+4]*V[.,4];
                VV[.,j]=(VV[.,j]+(1/(1+d))*q[t+1,(j-1)*5+5]*utilbeq(wdisbeq));

/* Now pick maximum value */;
              Vmax[i,j]=maxc(VV[.,j]);

            cell = indexcat(VV[.,j],Vmax[i,j]);

            Medicaid[i,j] = Medicaid[i,j] + (1/(1+r))*(q[t+1,(j-1)*5+1]*Medicaid2[cell,1] + q[t+1,(j-1)*5+2]*Medicaid2[
            insurance[i,j] = insurance[i,j] + (1/(1+r))*(q[t+1,(j-1)*5+1]*insurance2[cell,1] + q[t+1,(j-1)*5+2]*insuran
            Astream[((t-1)*wrow)+i,j] = indexcat(VV[.,j],Vmax[i,j]);
            Cstream[((t-1)*wrow)+i,j] = c[Astream[((t-1)*wrow)+i,j],1];

            j=j+1;
            endo;
        i=i+1;
        endo;
        V=Vmax;
    Medicaid2 = Medicaid;
    insurance2 = insurance;

    t=t-1;
    endo;

/* This gives you utility as of period 1.
Now go back to period 0 for healthy person. */;

    Vstar=(1/(1+rho))*((q[1,1])*V[.,1]+(q[1,2])*V[.,2]+(q[1,3])*V[.,3]+(q[1,4])*V[.,4]);
    Vstar=Vstar+(1/(1+d))*(q[1,5])*utilbeq(wdisbeq);

    Mstar=(1/(1+r))*((q[1,1])*Medicaid2[.,1]+(q[1,2])*Medicaid2[.,2]+(q[1,3])*Medicaid2[.,3]+(q[1,4])*Medicaid2[.,4]);
    Mstar=Mstar+(1/(1+d))*(q[1,5])*utilbeq(wdisbeq);

    Istar=(1/(1+r))*((q[1,1])*insurance2[.,1]+(q[1,2])*insurance2[.,2]+(q[1,3])*insurance2[.,3]+(q[1,4])*insurance2[.,4
    Istar=Istar+(1/(1+d))*(q[1,5])*utilbeq(wdisbeq);

/* Vstar is utility as of period 0 and given initial wealth for healthy person as of starting age */

if proctype==1;         /* proctype==1 means want to report optimal value only */

        wstart=w0*(1+r);
        wcompare=dotfge(wstart,wdis);
        wcrit=sumc(wcompare);
        if wcrit>=wrow;
```

```
                Vstar=Vstar[wrow];
            Mstar=Mstar[wrow];
            Istar=Istar[wrow];
        else;
        Vstar=Vstar[wcrit]+(Vstar[wcrit+1]-Vstar[wcrit])*((wstart-wdis[wcrit])/(wdis[wcrit+1]-wdis[wcrit]));
        Mstar=Mstar[wcrit]+(Mstar[wcrit+1]-Mstar[wcrit])*((wstart-wdis[wcrit])/(wdis[wcrit+1]-wdis[wcrit]));
        Istar=Istar[wcrit]+(Istar[wcrit+1]-Istar[wcrit])*((wstart-wdis[wcrit])/(wdis[wcrit+1]-wdis[wcrit]));
endif;
elseif proctype==2;         @this is the same as if proctype = 1, except we don't calculate Vstar, only Mstar@

        wstart=w0*(1+r);
        wcompare=dotfge(wstart,wdis);       @wcompare=(wstart.>=wdis);@
        wcrit=sumc(wcompare);
        if wcrit>=wrow;
        Mstar=Mstar[wrow];
        Istar=Istar[wrow];
        else;
        Mstar=Mstar[wcrit]+(Mstar[wcrit+1]-Mstar[wcrit])*((wstart-wdis[wcrit])/(wdis[wcrit+1]-wdis[wcrit]));
        Istar=Istar[wcrit]+(Istar[wcrit+1]-Istar[wcrit])*((wstart-wdis[wcrit])/(wdis[wcrit+1]-wdis[wcrit]));
endif;

endif;

/*SAVES THE ASSET AND CONSUMPTION MATRICES IF WE WANT TO GRAPH LATER*/
if graphs >0;
prec = 0;
field = 1;
fmat = "_w%*.*lf";

w = ftos(wcount,fmat,field,prec);

fmat = "_gam%*.*lf";

g = ftos(gam,fmat,field,prec);

fmat = "_grid%*.*lf";

gr = ftos(grid,fmat,field,prec);

fmat = "_MW%*.*lf";

money = ftos(MW*100, fmat, field, prec);

if proctype == 1;
    l="_own";
elseif proctype == 2;
    l="_none";
endif;

if male == 1;
    s="_M";
elseif male == 0;
    s="_F";
endif;
```

```
Amatrixname = "A" $+ s $+ w $+ g $+ gr $+ money $+ l;
cmatrixname = "C" $+ s $+ w $+ g $+ gr $+ money $+ l;

save ^Amatrixname = Astream;
save ^Cmatrixname = Cstream;
endif;

retp(Vstar, Mstar, Medicalstar, Istar);

endp;

/*=================================================================================*/
/*       END UTILITY PROCEDURE                                                     */
/*=================================================================================*/
```