```
> restart;
> ### This worksheet was written for Maple 16.01 Standard.
  ### May need tweaking for earlier versions of Maple or for Maple
  Classic.
  ### Last Revised 2012-10-01
  ### Report problems: contact@patricktoche.com
> ### Set display option
  mydisplayprecision:=3:
  interface(displayprecision=mydisplayprecision):
> ### Procedure to export plots

  MakePlot := proc(p::evaln, {[x,ext,extension]:=ps})
      local thename, theplace, opts:
      global N;
      thename := cat(convert(p,string),"_",convert(N,string),".",
  convert(x,string)):
      theplace := cat(currentdir(),kernelopts(dirsep),convert(N,
  string),kernelopts(dirsep)):
      if x = gif then
          opts := `color,portrait,noborder,transparent,height=512,
  width=512`: #default jpeg: height=360,width=480
      else
          #default gif : height=512,width=512
          opts := `color,portrait,noborder,transparent,height=360,
  width=480`:
      end if:
      plotsetup('x', 'plotoutput'=cat(theplace,thename),
  'plotoptions'=opts):
      print( plots:-display( eval(p), 'axesfont' = [ TIMES, 10 ],
  'labelfont' = [ TIMES, ROMAN, 10] ) ):
      plotsetup(default):
  end proc:

> ### Tractable Model Parameter Definitions
  ###   rho   : coefficient of relative risk aversion, CRRA
  ###   mu    : probability of job loss
  ###   R     : interest factor on financial wealth, i.e. R = 1+r
  ###   beta  : patience factor, i.e. inverse of discount factor
  ###   G     : growth factor of labor income
  ###   Gamma : Gamma = G/(1-mu)

> ############################## Incomplete
  ############################
  ### The Selection of Parameter Values is at the experimental
  stage ###
  ### Choices subject to change
   ###
  ### Not all figures have been tweaked or optimized
   ###
  ##################################################################
  ####

> ### Parameter values for ctdiscrete, fixing Gamma=1 (Zero Growth)
  ### To use this parameter configuration set N:=1;
  parameters[1] := [ R = 103/100, beta = 100/110, Gamma = 1 ]:
      'parameters[1]' = evalf(%);
      'R*beta' = evalf(eval(R*beta,parameters[1]));
```

$$parameters_1 = \left[R = 1.03,\ \beta = 0.909,\ \Gamma = 1.\right]$$

$$R\,\beta = 0.936 \tag{1}$$

```
> ### Parameter values for ctdiscrete, fixing G=1 (Zero Growth)
  ### To use this parameter configuration set N:=2;

  parameters[2] := [ R = 103/100, beta = 100/110, Gamma = 1/(1-mu)
  ]:
      'parameters[2]' = evalf(%);
      'R*beta' = evalf(eval(R*beta,parameters[2]));
```

$$parameters_2 = \left[R = 1.03,\ \beta = 0.909,\ \Gamma = \frac{1}{1-\mu}\right]$$

$$R\,\beta = 0.936 \tag{2}$$

```
> ### Parameter values from cssUSsaving, 16 March 2012, section 5.2
  ### To use this parameter configuration set N:=3;
  ### R=1.04 and beta=0.975=10000/10256,e at annual frequency.
  ### R=1.01 and beta=1-0.0064=0.994, at quarterly frequency

  parameters[3] := [ R = 104/100, beta = 10000/10256, Gamma =
  101/100/(1-mu) ]:
      'parameters[3]' = evalf(%);
      'R*beta' = evalf(eval(R*beta,parameters[3]));
```

$$parameters_3 = \left[R = 1.04,\ \beta = 0.975,\ \Gamma = \frac{1.01}{1-\mu}\right]$$

$$R\,\beta = 1.01 \tag{3}$$

```
> ### Parameter values, fixing Gamma=101/100 (Positive Growth)
  ### To use this parameter configuration set N:=4;

  parameters[4] := [ R = 103/100, beta = 100/110, Gamma = 101/100 ]
  :
      'parameters[4]' = evalf(%);
      'R*beta' = evalf(eval(R*beta,parameters[4]));
```

$$parameters_4 = \left[R = 1.03,\ \beta = 0.909,\ \Gamma = 1.01\right]$$

$$R\,\beta = 0.936 \tag{4}$$

```
> ### Parameter values, fixing Gamma=101/100  (Positive Growth, R*
  beta=1)
  ### To use this parameter configuration set N:=5;

  parameters[5] := [ R = 103/100, beta = 100/103, Gamma = 101/100 ]
  :
      'parameters[5]' = evalf(%);
      'R*beta' = evalf(eval(R*beta,parameters[5]));
```

$$parameters_5 = \left[R = 1.03,\ \beta = 0.971,\ \Gamma = 1.01\right]$$

$$R\,\beta=1. \tag{5}$$

```
> ### Set parameter values from the configurations above
  ### Select a value for N below, save, and  Edit -> Execute ->
  Worksheet

  N := 1:  # Parameter lists are numbered: N = 1,2,3...
      params := parameters[N]:
      'params' = evalf(params);
```

$$params=\left[R=1.03,\,\beta=0.909,\,\Gamma=1.\right] \tag{6}$$

```
> ### Store selected individual parameters for convenience

  Rf := subs(params,R):
  betaf := subs(params,beta):
  Gammaf := subs(params,Gamma):
```

```
> ### Marginal propensity to consume in unemployment

  mpcu := (R,beta,rho) -> 1-(R*beta)^(1/rho)/R:
      'mpcu' = mpcu(R,beta,rho);
```

$$mpcu=1-\frac{(R\,\beta)^{\frac{1}{\rho}}}{R} \tag{7}$$

```
> ### Target wealth-income ratio

  m := (R,beta,Gamma,rho,mu) -> 1 + 1 / ( Gamma/R - 1 + mpcu(R,
  beta,rho) * ( 1 + ( ((R*beta)^(1/rho)/Gamma)^(-rho)-1 ) / mu )^
  (1/rho) ):
      'm' = m(R,beta,Gamma,rho,mu);
```

$$m=1+\frac{1}{\frac{\Gamma}{R}-1+\left(1-\frac{(R\,\beta)^{\frac{1}{\rho}}}{R}\right)\left(1+\frac{\left(\frac{(R\,\beta)^{\frac{1}{\rho}}}{\Gamma}\right)^{-\rho}-1}{\mu}\right)^{\frac{1}{\rho}}} \tag{8}$$

```
> ### Target saving rate
  ### from pi/(1-pi)=rhs (c.f. equation in the text), we have pi=
  rhs/(1+rhs), so we have s=1-pi=1/(1+rhs)

  s := (R,beta,Gamma,rho,mu) -> 1 / (1 + mpcu(R,beta,rho)*(R/Gamma)
  *((((R*beta)^(1/rho)/Gamma)^(-rho)-(1-mu))/mu)^(1/rho) ):
      's' = s(R,beta,Gamma,rho,mu);
```

(9)

$$s=\frac{1}{1+\frac{\left(1-\frac{(R\beta)^{\frac{1}{\rho}}}{R}\right)R\left(\frac{\left(\frac{(R\beta)^{\frac{1}{\rho}}}{\Gamma}\right)^{-\rho}-1+\mu}{\mu}\right)^{\frac{1}{\rho}}}{\Gamma}} \qquad \textbf{(9)}$$

```
> ### Create a list of values for rho

  rholist := [ seq(k, k = 1 .. 20) ]:
      'rho' = rholist[1..10];
```

$$\rho=[1,2,3,4,5,6,7,8,9,10] \qquad \textbf{(10)}$$

```
> ### Create a list of values for mu

  mulist := [ 0, seq(2^k/100, k = 0 .. 20) ]:
      'mu' = evalf(%)[1..10];
```

$$\mu=[0.,0.0100,0.0200,0.0400,0.0800,0.160,0.320,0.640,1.28,2.56] \qquad \textbf{(11)}$$

```
> ### Check RIC and GIC Conditions

  RIC := (R,beta,rho) -> (R*beta)^(1/rho)/R:
  RICf := rho -> RIC(subs(params,R),subs(params,beta),rho):
  GIC := (R,beta,rho,Gamma) -> (R*beta)^(1/rho)/Gamma:
  GICf := (rho,mu) -> GIC(subs(params,R),subs(params,beta),rho,subs
  (params,Gamma)):

  ### Check the RIC
  Matrix([seq( [seq( is(RICf(rho)<1), mu=mulist[2..8])],rho=rholist
  [1..10])]):
      LinearAlgebra:-Transpose(%);

  ### Check the GIC
  Matrix([seq( [seq( is(GICf(rho,mu)<1), mu=mulist[2..8])],rho=
  rholist[1..10])]):
      LinearAlgebra:-Transpose(%);

  ### Check the strong GIC
  Matrix([seq( [seq( is(GICf(rho,mu)<(1-mu)^(-1/rho)), mu=mulist[2.
  .8])],rho=rholist[1..10])]):
      LinearAlgebra:-Transpose(%);
```

$$
\begin{bmatrix}
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true}
\end{bmatrix}
$$

$$
\begin{bmatrix}
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true}
\end{bmatrix}
$$

$$
\begin{bmatrix}
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} \\
\mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true} & \mathit{true}
\end{bmatrix} \tag{12}
$$

```
> ### Target wealth-income ratio for fixed values of R,Gamma,beta

  eval(m(R,beta,Gamma,rho,mu),params):
  mf := unapply(%,(rho,mu)):
  interface(displayprecision=3):
      'm' = evalf(mf(rho,mu));
  interface(displayprecision=mydisplayprecision):
```

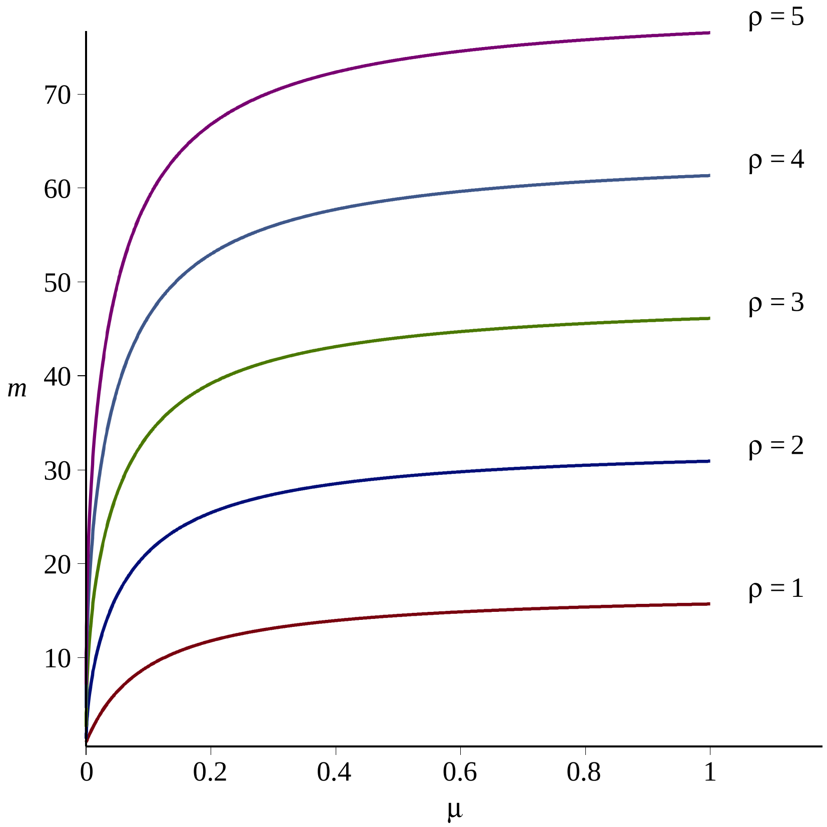
$$
m = 1 + \frac{1}{-0.0291 + \left(1 - 0.971\ 0.936^{\frac{1}{\rho}}\right) \left(1 + \frac{\left(0.936^{\frac{1}{\rho}}\right)^{-\rho} - 1}{\mu}\right)^{\frac{1}{\rho}}} \tag{13}
$$

```
> ### Plot of m as rho and mu vary

  mTargetUrateVariesCRRAVaries := plots:-display( plot3d(mf(rho,
  mu), rho = 1..5, mu = 0..1)
      , 'axes' = normal
      , 'style' = surfacecontour
      , 'shading' = zhue
      , 'lightmodel' = light1
      , 'tickmarks' = [ 6, 6, 4 ]
      , 'labels' = [ rho, mu, 'm' ]
      , 'view' = [ 1 .. 5, 0 .. 1, default ]
      , 'orientation' = [ -10, 50 ]
    ) :  # % ;
```

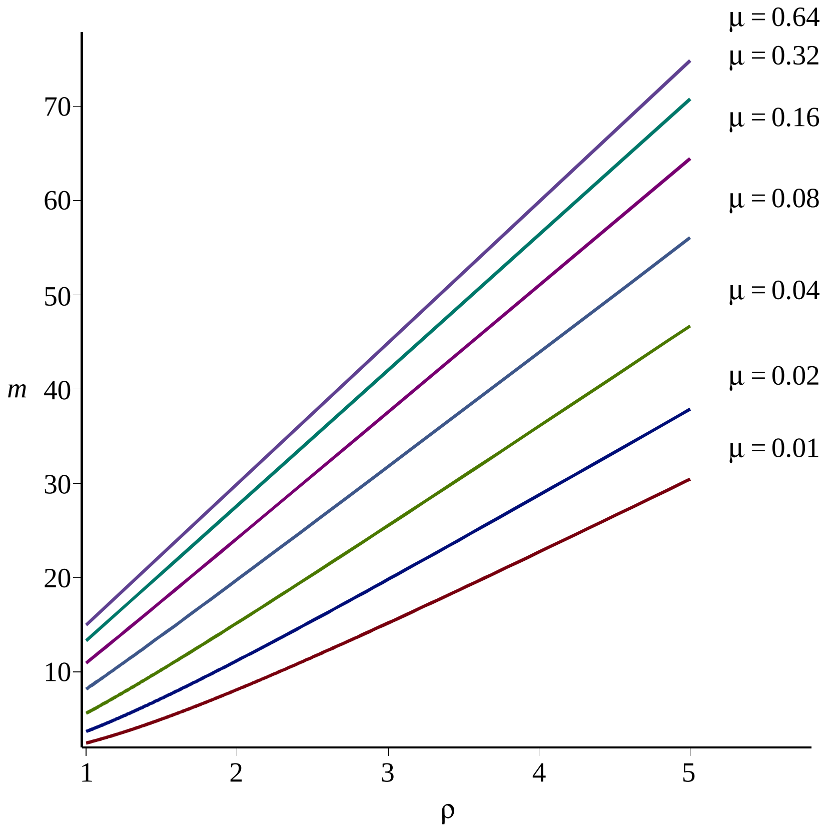
```
> ### Animated plot of m as rho and mu vary

  mTargetUrateVariesCRRAVariesAnimation :=  plots:-display(
  mTargetUrateVariesCRRAVaries
      , 'viewpoint' = ["circleright", frames=200]
    ) : # % ;
```

```
> ### Set position of the plot labels, tweaked for stated parameter
  values

  if N=2 then
      xmu:=rho->0.2/rho:  ymu:=rho->1.4*mf(rho,xmu(rho)): # fix x-
  value, vary y-value
      xrho:=mu->5.2:      yrho:=mu->mf(xrho(mu),mu):  # fix x-
  value, vary y-value
  else
      xmu:=rho->1.05: ymu:=rho->mf(rho,xmu(rho)): # fix x-value,
  vary y-value
      xrho:=mu->5.2:  yrho:=mu->mf(xrho(mu),mu):  # fix x-value,
  vary y-value
  end if:
```

```
> ### Plot of m as mu varies for fixed values of rho

  plot_m_mu := plot( [ seq( mf(rho,mu) , rho=rholist[1..5] ) ]
      , mu = 0 .. 1
      , 'numpoints' = 1000
      , 'tickmarks' = [ 6, 6 ]
      , 'labels' = [ mu, 'm' ]
  #    , 'legend' = [ seq( 'rho' = k, k = rholist[1..5] ) ]
  #    , 'legendstyle' = [ 'font' = [TIMES,ROMAN,8], 'location' =
  bottom ]
      , 'view' = [ 0 .. 1.18, default ]
    ) :

  #### plot labels

  ptxt := seq( plots:-textplot([xmu(rho),ymu(rho),'typeset'('rho',
  " = ", rho)], 'align'={'above','right'}), rho=rholist[1..5]):

  mTargetCRRAFixedUrateVaries := plots:-display([plot_m_mu,ptxt]):
  %;
```



```
> ### Plot of m as rho varies for fixed values of mu

  interface(displayprecision=2):

  plot_m_rho := plot( [seq(mf(rho,mu),mu=mulist[2..8])]
      , rho = 1 .. 5
      , 'numpoints' = 1000
      , 'tickmarks' = [ 6, 6 ]
      , 'labels' = [ rho, 'm' ]
  #    , 'legend' = [ seq( 'mu' = evalf(k), k = mulist[2..8] ) ]
  #    , 'legendstyle' = [ 'font' = [TIMES,ROMAN,8], 'location' =
  bottom ]
      , 'view' = [ 1 .. 5.8, default ]
    ) :

  #### plot labels

  ptxt := seq( plots:-textplot([xrho(mu),yrho(mu),'typeset'('mu', "
  = ", evalf(mu))], 'align'={'above','right'}), mu=mulist[2..8]):

  mTargetUrateFixedCRRAVaries := plots:-display([plot_m_rho,ptxt]):
  %;
  interface(displayprecision=mydisplayprecision):
```



```
> ### Table of target values m as rho and mu run through lists

  interface(displayprecision=6):
  mvalues := Matrix([seq( [seq(mf(rho,mu), rho=rholist[1..8])],mu=
  mulist[2..8])]):
  mvalues := ArrayTools:-Concatenate(2,Vector[column](evalf[2]
  (mulist[2..8])),mvalues):
  mvalues := ArrayTools:-Concatenate(1,Vector[row]([0,op(rholist[1.
  .8])]),mvalues):
      'mvalues' = evalf(%);
  interface(displayprecision=mydisplayprecision):
```

mvalues **(14)**

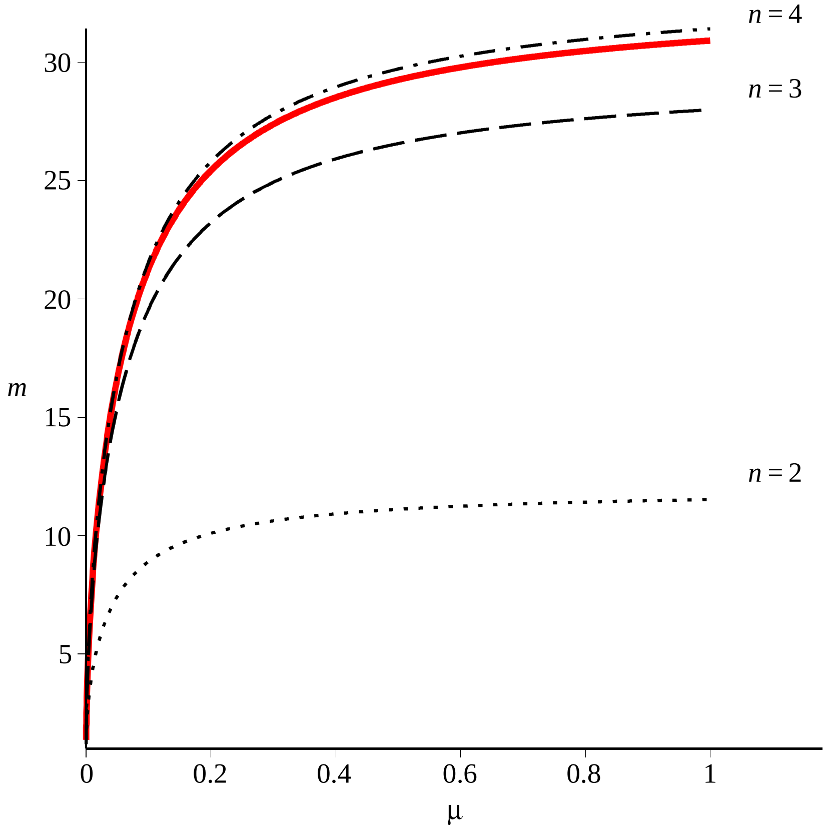
$$= \begin{bmatrix} 0. & 1. & 2. & 3. & 4. & 5. & 6. & 7. & 8. \\ 0.010 & 2.47143 & 8.14950 & 15.2124 & 22.7429 & 30.4817 & 38.3307 & 46.2449 & 54.2009 \\ 0.020 & 3.69762 & 11.2244 & 19.8402 & 28.7968 & 37.9017 & 47.0842 & 56.3122 & 65.5693 \\ 0.040 & 5.62449 & 15.2229 & 25.5514 & 36.0945 & 46.7290 & 57.4107 & 68.1200 & 78.8468 \\ 0.080 & 8.19365 & 19.8018 & 31.8058 & 43.9202 & 56.0806 & 68.2646 & 80.4623 & 92.6687 \\ 0.16 & 10.9604 & 24.1917 & 37.5882 & 51.0288 & 64.4875 & 77.9554 & 91.4286 & 104.905 \\ 0.32 & 13.3320 & 27.6545 & 42.0277 & 56.4140 & 70.8057 & 85.2000 & 99.5959 & 113.993 \\ 0.64 & 14.9985 & 29.9589 & 44.9293 & 59.9023 & 74.8762 & 89.8507 & 104.825 & 119.800 \end{bmatrix}$$

```
> ### Check of the accuracy of various approximations
  ### The plot shows that n>3 is needed for decent approximation

  Rho := 2: # Fix a value of rho = Rho

  mfn := (rho,mu,n) -> evalf[n](mf(rho,mu)):
      'mfn' = [mfn(Rho,mu,1),mfn(Rho,mu,2),mfn(Rho,mu,3),mfn(Rho,
  mu,4),mfn(Rho,mu,5)];

  plot_mff_mu := plot( mf(Rho,mu)
      , mu = 0 .. 1
      , 'numpoints' = 1000
      , 'color' = red
      , 'thickness' = 3
      , 'linestyle' = solid
    ) :

  plot_mfn_mu := n -> plot( mfn(Rho,mu,n)
      , mu = 0 .. 1
      , 'numpoints' = 1000
      , 'color' = black
      , 'thickness' = 1
      , 'linestyle' = n
    ) :


  ### plot labels
  xmu:=n->1.05: ymu:=n->mfn(Rho,1,n): # fix x-value, vary y-value
  ptxt := seq( plots:-textplot([xmu(n),ymu(n),'typeset'('n', " = ",
  n)], 'align'={'above','right'}), n=2..4):

  mTargetCRRAFixedUrateVariesApproximations :=
      plots:-display([plot_mff_mu,plot_mfn_mu(2),plot_mfn_mu(3),
  plot_mfn_mu(4),ptxt]
          , 'tickmarks' = [ 6, 6 ]
          , 'labels' = [ mu, 'm' ]
          , 'view' = [ 0 .. 1.18, default ]
        ) : %;
```

$$mfn = \left[1 + \frac{1}{-0.03 + 0.1\sqrt{1 + \frac{0.07}{\mu}}}, 1 + \frac{1}{-0.029 + 0.12\sqrt{1 + \frac{0.068}{\mu}}}, 1\right.$$

$$+\frac{1}{-0.0291+0.064\sqrt{1+\frac{0.0680}{\mu}}}, 1+\frac{1}{-0.0291+0.0600\sqrt{1+\frac{0.0680}{\mu}}}, 1$$

$$+\frac{1}{-0.0291+0.0606\sqrt{1+\frac{0.0680}{\mu}}}\Bigg]$$



```
> ##########################
> ### Asymptotic values of m as risk-aversion rho becomes
  arbitrarily large

  asymptotic_m_mu := [seq(limit(mf(rho,mu),rho=infinity), mu=mulist
  [2..20])];
```

$$asymptotic\_m\_mu := [\infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty, \infty] \tag{15}$$

```
> ### Derivative of m with respect to R

  dm := (R,beta,Gamma,rho,mu) -> diff(m(R,beta,Gamma,rho,mu),R):
  eval(dm(R,beta,Gamma,rho,mu),params):
  dmf := unapply(%,(rho,mu)):
  interface(displayprecision=4):
    'dm' = evalf(dmf(rho,mu));
interface(displayprecision=mydisplayprecision):
```

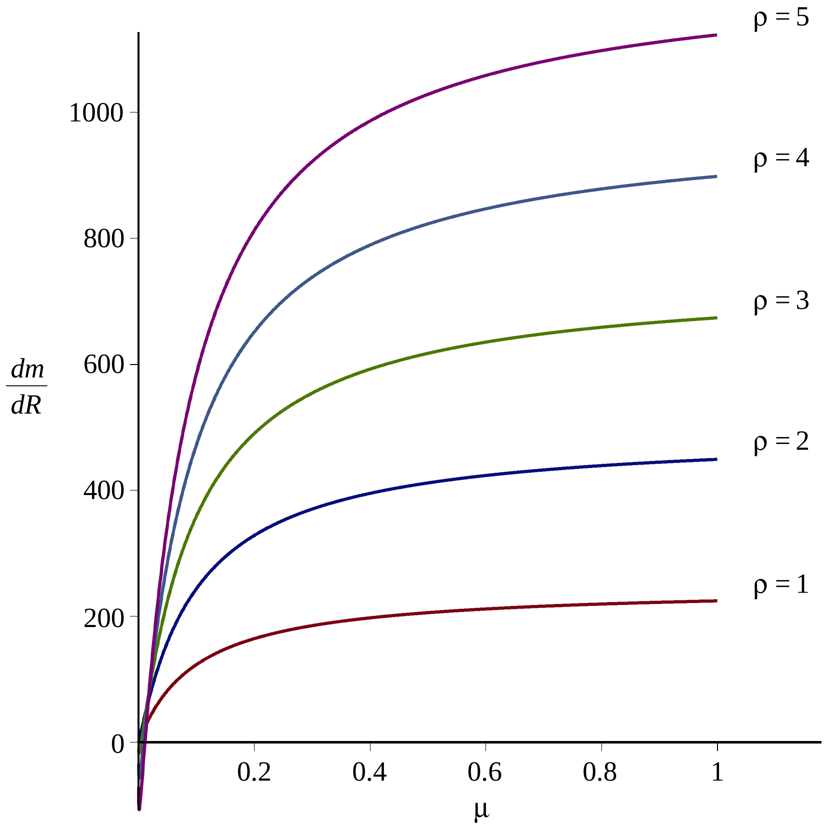
$$dm = -\left(\left(-0.9426 + \left(-\frac{0.9426\, 0.9364^{\frac{1}{\rho}}}{\rho} + 0.9426\, 0.9364^{\frac{1}{\rho}} \right) \left(1 + \frac{\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1}{\mu} \right)^{\frac{1}{\rho}} \right. \right.$$

$$\left. - \frac{0.9709 \left(1 - 0.9709\, 0.9364^{\frac{1}{\rho}}\right) \left(1 + \frac{\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1}{\mu} \right)^{\frac{1}{\rho}} \left(0.9364^{\frac{1}{\rho}}\right)^{-\rho}}{\rho\, \mu \left(1 + \frac{\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1}{\mu} \right)} \right)$$

$$\left/ \left(-0.02913 + \left(1 - 0.9709\, 0.9364^{\frac{1}{\rho}}\right) \left(1 + \frac{\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1}{\mu} \right)^{\frac{1}{\rho}} \right)^{2} \right. \qquad \textbf{(16)}$$

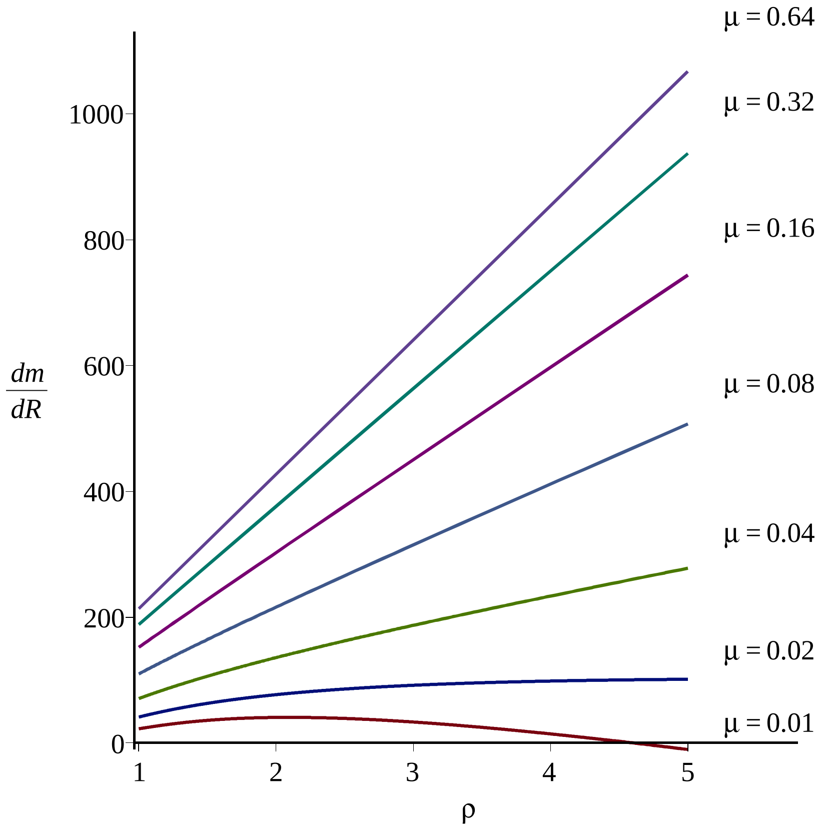
```
> ### Set position of the plot labels, tweaked for stated parameter
values

if N=2 then
    xmu:=rho->0.12:  ymu:=rho->-4+1.6*dmf(rho,xmu(rho)): # fix x-
value, vary y-value
    xrho:=mu->5.2:      yrho:=mu->dmf(xrho(mu),mu):  # fix x-
value, vary y-value
else
    xmu:=rho->1.05: ymu:=rho->dmf(rho,xmu(rho)): # fix x-value,
vary y-value
    xrho:=mu->5.2:  yrho:=mu->dmf(xrho(mu),mu)+20:  # fix x-
value, vary y-value
end if:
```

```
> ### Plot of derivative of m with respect to R, for fixed values
of rho

plot_dmdR_mu := plot( [ seq( dmf(rho,mu) , rho=rholist[1..5] ) ]
    , mu = 0 .. 1
    , 'numpoints' = 1000
    , 'tickmarks' = [ 6, 6 ]
    , 'labels' = [ mu, 'dm/dR' ]
    , 'view' = [ 0 .. 1.18, default ]
 ) :

#### plot labels

ptxt := seq( plots:-textplot([xmu(rho),ymu(rho),'typeset'('rho',
" = ", rho)], 'align'={'above','right'}), rho=rholist[1..5]):

if N = 2 then
    theview := [ 0 .. 1, -10 .. 28 ] :
else
    theview := default :
end if:

mSlopeCRRAFixedUrateVaries := plots:-display( [plot_dmdR_mu,
ptxt], 'view' = theview ): %;
```



```
> ### Plot of derivative of m with respect to R, for fixed values
  of mu

  interface(displayprecision=2):
```

```
plot_dmdR_rho := plot( [ seq( dmf(rho,mu) , mu=mulist[2..8] ) ]
    , rho = 1 .. 5
    , 'numpoints' = 1000
    , 'tickmarks' = [ 6, 6 ]
    , 'labels' = [ rho, 'dm/dR' ]
    , 'view' = [ 1 .. 5.8, default ]
  ) :

#### plot labels

ptxt := seq( plots:-textplot([xrho(mu),yrho(mu),'typeset'('mu', "
= ", evalf(mu))], 'align'={'above','right'}), mu=mulist[2..8]):

mSlopeUrateFixedCRRAVaries := plots:-display([plot_dmdR_rho,ptxt]
): %;

interface(displayprecision=mydisplayprecision):
```



```
> ### Table of percentage change in target values m after 1% Change
  in After-Tax Interest Rate
  ### Mid-Point Formula

  interface(displayprecision=6):
```

```
mchanges := Matrix([seq( [seq( 100*(m(Rf,betaf,Gammaf,rho,mu)-m
(Rf-1/100,betaf,Gammaf,rho,mu))/((m(Rf,betaf,Gammaf,rho,mu)+m
(Rf-1/100,betaf,Gammaf,rho,mu))/2) ,rho=rholist[1..8] )],mu=
mulist[2..8])]):
mchanges := ArrayTools:-Concatenate(2,Vector[column](evalf[2]
(mulist[2..8])),mchanges):
mchanges := ArrayTools:-Concatenate(1,Vector[row]([0,op(rholist
[1..8])]),mchanges):
    'mchanges' = evalf(%);
interface(displayprecision=mydisplayprecision):
```

```
mchanges = [[0., 1., 2., 3., 4., 5., 6., 7., 8.],                                              (17)
   [0.010, 8.27690, 4.18009, 1.27797, −0.359146, −1.38644, −2.08698, −2.59413,
   −2.97785],
   [0.020, 10.2378, 5.98863, 3.70794, 2.45720, 1.67806, 1.14811, 0.764863, 0.474995],
   [0.040, 11.6134, 8.01884, 6.39410, 5.52185, 4.98188, 4.61553, 4.35092, 4.15090],
   [0.080, 12.4498, 9.97832, 8.96384, 8.42685, 8.09593, 7.87184, 7.71015, 7.58799],
   [0.16, 12.9149, 11.5658, 11.0369, 10.7589, 10.5879, 10.4722, 10.3888, 10.3258],
   [0.32, 13.1608, 12.6550, 12.4570, 12.3527, 12.2883, 12.2447, 12.2132, 12.1894],
   [0.64, 13.2872, 13.3121, 13.3135, 13.3128, 13.3120, 13.3113, 13.3107, 13.3101]]
```

```
> ##########################
> ### Target saving rate for fixed values of R,Gamma,beta

  eval(s(R,beta,Gamma,rho,mu),params):
  sf := unapply(%,(rho,mu)):
  interface(displayprecision=4):
      's' = evalf(sf(rho,mu));
  interface(displayprecision=mydisplayprecision):
```

$$s = \frac{1}{1 + 1.030\left(1 - 0.9709\, 0.9364^{\frac{1}{\rho}}\right)\left(\frac{\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1 + \mu}{\mu}\right)^{\frac{1}{\rho}}} \qquad \textbf{(18)}$$

```
> ### Plot of s as rho and mu vary

  sTargetUrateVariesCRRAVaries := plots:-display( plot3d(sf(rho,
  mu), rho = 1..5, mu = 0..1)
      , 'axes' = normal
      , 'style' = surfacecontour
      , 'shading' = zhue
      , 'lightmodel' = light1
      , 'tickmarks' = [ 6, 6, 4 ]
      , 'labels' = [ rho, mu, 's' ]
      , 'view' = [ 1 .. 5, 0 .. 1, 0.5 .. 1 ]
      , 'orientation' = [ -10, 50 ]
    ) :

  plot_s_rho_mu;
```

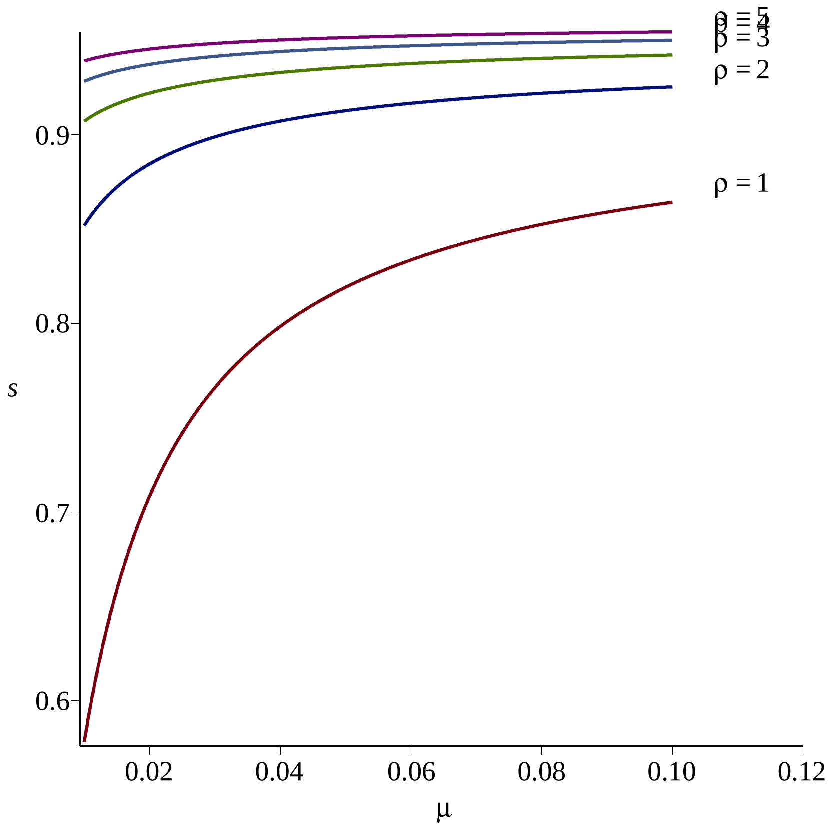
(19)

plot_s_rho_mu **(19)**

```
> ### Animated plot of m as rho and mu vary

  sTargetUrateVariesCRRAVariesAnimation :=  plots:-display(
  sTargetUrateVariesCRRAVaries
      , 'viewpoint' = ["circleright", frames=200]
    ) : # % ;
```

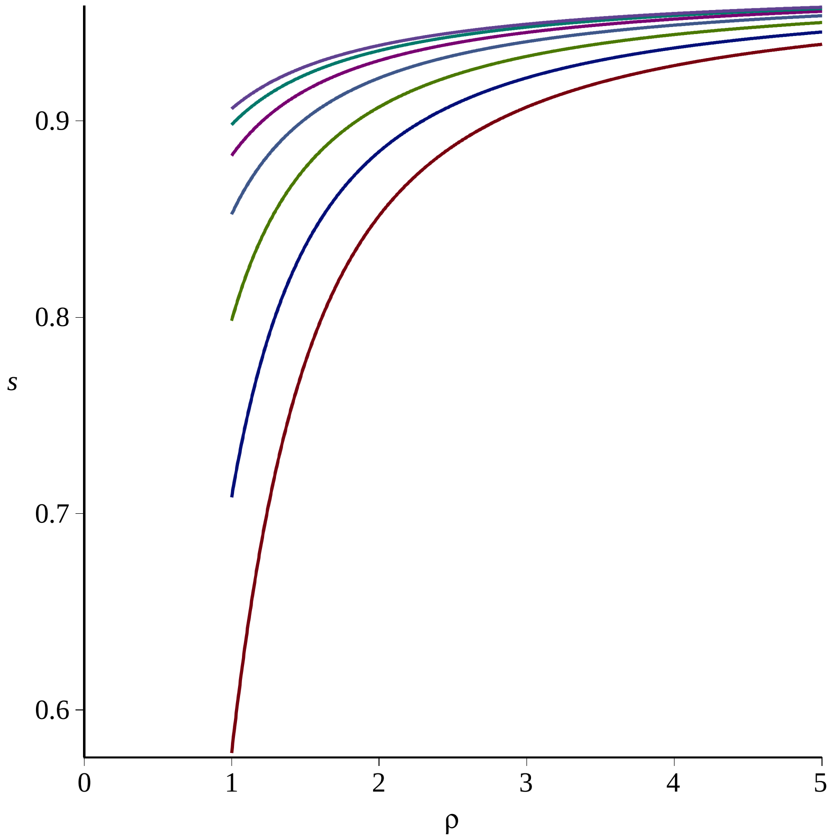
```
> ### Set position of the plot labels, tweaked for stated parameter
  values

  mumin := 0.01:
  mumax := 0.1:
  rhomin := 1:
  rhomax := 5:

  if N=2 then
      xmu:=rho->0.2/rho:       ymu:=rho->1.4*sf(rho,xmu(rho)): # fix
  x-value, vary y-value
      xrho:=mu->1.05*rhomax:  yrho:=mu->sf(xrho(mu),mu):  # fix x-
  value, vary y-value
  elif N=4 or N=5 then
      xmu:=rho->1.05*mumax:   ymu:=rho->sf(rho,xmu(rho)): # fix x-
  value, vary y-value
      xrho:=mu->1:            yrho:=mu->sf(xrho(mu),mu):  # fix x-
  value, vary y-value
  else
      xmu:=rho->1.05*mumax:   ymu:=rho->sf(rho,xmu(rho)): # fix x-
  value, vary y-value
      xrho:=mu->1.05*rhomax:  yrho:=mu->sf(xrho(mu),mu):  # fix x-
  value, vary y-value
  end if:
```

```
> ### Plot of s as mu varies for fixed values of rho

  plot_s_mu := plot( [ seq( sf(rho,mu) , rho=rholist[1..rhomax] ) ]
      , mu = mumin .. mumax
      , 'numpoints' = 1000
      , 'tickmarks' = [ 6, 6 ]
      , 'labels' = [ mu, 's' ]
  #    , 'legend' = [ seq( 'rho' = k, k = rholist[rhomin..rhomax] )
  ]
  #    , 'legendstyle' = [ 'font' = [TIMES,ROMAN,8], 'location' =
  bottom ]
  #    , 'view' = [ mumin .. 1.2*mumax, 0.85 .. max([seq(evalf(sf
  (rho,mumax)),rho=rholist[rhomin..rhomax]])]) ]
      , 'view' = [ mumin .. 1.2*mumax
          , min([seq(evalf(sf(rho,mumin)),rho=rholist[rhomin..
  rhomax])]) .. max([seq(evalf(sf(rho,mumax)),rho=rholist[rhomin..
  rhomax])]) ]
    ) :

  #### plot labels

  ptxt := seq( plots:-textplot([xmu(rho),ymu(rho),'typeset'('rho',
  " = ", rho)], 'align'={'above','right'}), rho=rholist[rhomin..
rhomax]):

sTargetCRRAFixedUrateVaries := plots:-display([plot_s_mu,ptxt]):
%;
```



```
> ### Plot of s as rho varies for fixed values of mu

  interface(displayprecision=2):

  plot_s_rho := plot( [seq(sf(rho,mu),mu=mulist[2..8])]
      , rho = 1 .. 5
      , 'numpoints' = 1000
      , 'tickmarks' = [ 6, 6 ]
      , 'labels' = [ rho, 's' ]
  #     , 'legend' = [ seq( 'mu' = evalf(k), k = mulist[2..8] ) ]
  #     , 'legendstyle' = [ 'font' = [TIMES,ROMAN,8], 'location' =
  bottom ]
      , 'view' = [ 0 .. 5, default ]
    ) :

  #### plot labels

  if N=4 or N=5 then  # specifically tweaked for parameter values
```

```
N=4
    ptxt := seq( plots:-textplot([xrho(mu)-0.9,yrho(mu),'typeset'
('mu', " = ", evalf(mu))], 'align'={'above','right'}), mu=mulist
[2..8]):
else
    ptxt := seq( plots:-textplot([xrho(mu),yrho(mu),'typeset'
('mu', " = ", evalf(mu))], 'align'={'above','right'}), mu=mulist
[2..8]):
end if:

sTargetUrateFixedCRRAVaries := plots:-display([plot_s_rho,ptxt]):
%;

interface(displayprecision=mydisplayprecision):
```



```
> ### Table of target values s as rho and mu run through lists

  interface(displayprecision=6):
  svalues := Matrix([seq( [seq(sf(rho,mu), rho=rholist[1..8])],mu=
  mulist[2..8])]):
  svalues := ArrayTools:-Concatenate(2,Vector[column](evalf[2]
  (mulist[2..8])),svalues):
  svalues := ArrayTools:-Concatenate(1,Vector[row]([0,op(rholist[1.
.8])]),svalues):
    'svalues' = evalf(%);
interface(displayprecision=mydisplayprecision):
```

svalues **(20)**

= [[0., 1., 2., 3., 4., 5., 6., 7., 8.],
[0.010, 0.578035, 0.851741, 0.907053, 0.928185, 0.939023, 0.945545, 0.949880, 0.952961
],
[0.020, 0.708307, 0.884377, 0.921939, 0.937159, 0.945258, 0.950254, 0.953633, 0.956067
],
[0.040, 0.798258, 0.907096, 0.932877, 0.943976, 0.950097, 0.953963, 0.956621, 0.958560
],
[0.080, 0.852383, 0.921844, 0.940349, 0.948768, 0.953562, 0.956652, 0.958808, 0.960397
],
[0.16, 0.882294, 0.930741, 0.945045, 0.951848, 0.955819, 0.958420, 0.960255, 0.961619],
[0.32, 0.898051, 0.935767, 0.947773, 0.953664, 0.957162, 0.959479, 0.961126, 0.962357],
[0.64, 0.906142, 0.938467, 0.949265, 0.954666, 0.957907, 0.960068, 0.961612, 0.962770]]

```
> ### Elasticity of s with respect to R

  ds := (R,beta,Gamma,rho,mu) -> diff(s(R,beta,Gamma,rho,mu),R):
  es :=  (R,beta,Gamma,rho,mu) -> R*ds(R,beta,Gamma,rho,mu)/s(R,
  beta,Gamma,rho,mu):
  eval(es(R,beta,Gamma,rho,mu),params):
  esf := unapply(%,(rho,mu)):
  interface(displayprecision=4):
      'es' = evalf(esf(rho,mu));
  interface(displayprecision=mydisplayprecision):
```

$$es = -\left(1.030\left(1.030\left(-\frac{0.9426\, 0.9364^{\frac{1}{\rho}}}{\rho} + 0.9426\, 0.9364^{\frac{1}{\rho}}\right)\left(\frac{\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1 + \mu}{\mu}\right)^{\frac{1}{\rho}}\right.\right. \quad \textbf{(21)}$$

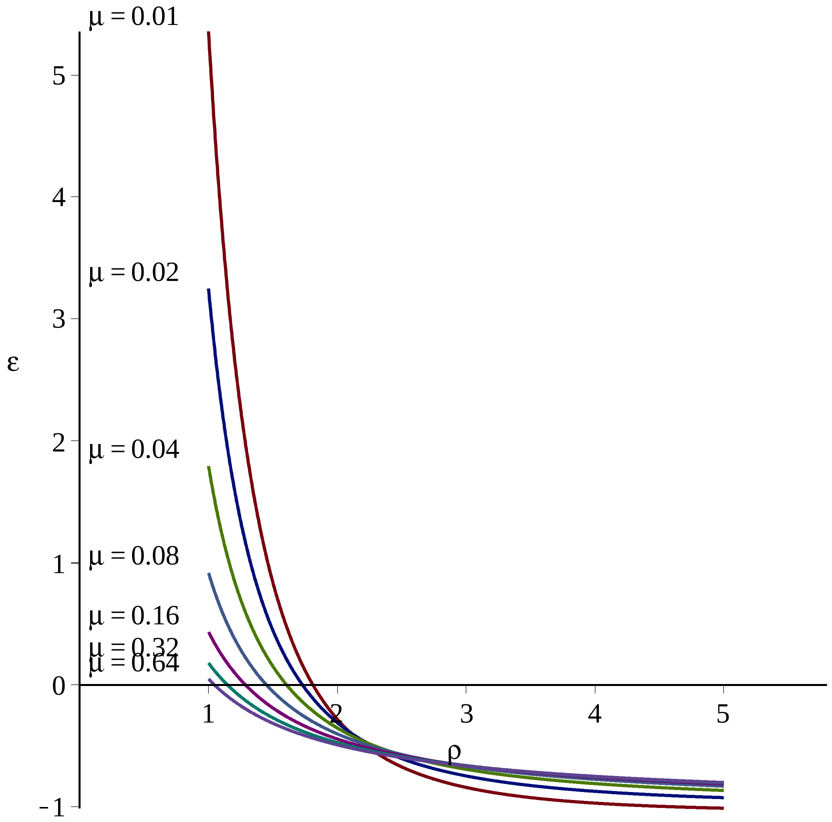
$$\left. + \left(1 - 0.9709\, 0.9364^{\frac{1}{\rho}}\right)\left(\frac{\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1 + \mu}{\mu}\right)^{\frac{1}{\rho}}\right.$$

$$- \frac{\left(1 - 0.9709\; 0.9364^{\frac{1}{\rho}}\right) \left(\frac{\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1 + \mu}{\mu}\right)^{\frac{1}{\rho}} \left(0.9364^{\frac{1}{\rho}}\right)^{-\rho}}{\rho \left(\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1 + \mu\right)} \Bigg)\Bigg) \Bigg/ \Bigg(1$$

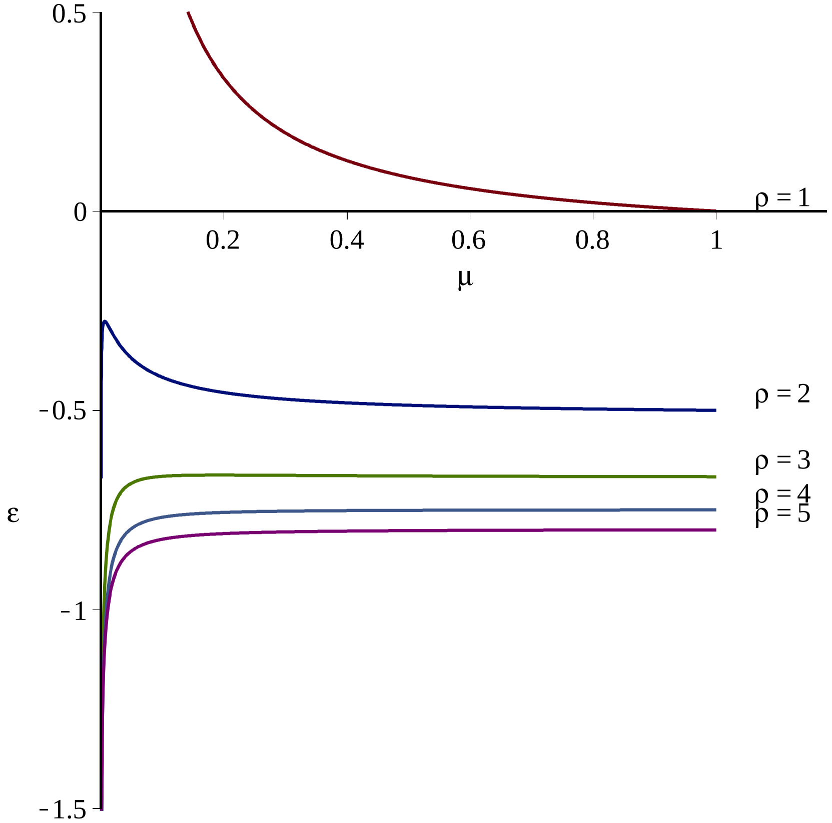
$$+ 1.030 \left(1 - 0.9709\; 0.9364^{\frac{1}{\rho}}\right) \left(\frac{\left(0.9364^{\frac{1}{\rho}}\right)^{-\rho} - 1 + \mu}{\mu}\right)^{\frac{1}{\rho}} \Bigg)$$

```
> ### Set position of the plot labels, tweaked for stated parameter
  values

  mumin := 1.0:
  mumax := 1.0:
  rhomin := 1:
  rhomax := 5:

  xmu:=rho->1.05*mumax:   ymu:=rho->esf(rho,xmu(rho)): # fix x-
  value, vary y-value
  xrho:=mu->mumin:        yrho:=mu->esf(xrho(mu),mu):  # fix x-
  value, vary y-value
```

```
> ### Plot of the elasticity of s with respect to R, for fixed
  values of mu

  interface(displayprecision=2):

  plot_es_rho := plot( [ seq( esf(rho,mu) , mu=mulist[2..8] ) ]
      , rho = 1 .. 5
      , 'numpoints' = 1000
      , 'tickmarks' = [ 6, 6 ]
      , 'labels' = [ rho, epsilon ]
      , 'view' = [ 0 .. 5.8, default ]
    ) :

  #### plot labels

  ptxt := seq( plots:-textplot([xrho(mu)-1,yrho(mu),'typeset'('mu',
  " = ", evalf(mu))], 'align'={'above','right'}), mu=mulist[2..8]):

  sElasticityUrateFixedCRRAVaries := plots:-display([plot_es_rho,
  ptxt]): %;

  interface(displayprecision=mydisplayprecision):
```



```
> ### Plot of the elasticity of s with respect to R, for fixed
  values of rho

  plot_es_mu := plot( [ seq( esf(rho,mu) , rho=rholist[1..5] ) ]
      , mu = 0 .. 1
      , 'numpoints' = 1000
      , 'tickmarks' = [ 6, 6 ]
      , 'labels' = [ mu, epsilon ]
      , 'view' = [ 0 .. 1.18, default ]
    ) :

  #### plot labels

  ptxt := seq( plots:-textplot([xmu(rho),ymu(rho),'typeset'('rho',
  " = ", rho)], 'align'={'above','right'}), rho=rholist[1..5]):

  sElasticityCRRAFixedUrateVaries := plots:-display([plot_es_mu,
  ptxt], 'view' = [ default, -3/2 .. 1/2 ]): %;
```



```
> ### Table of elasticity of target saving rate s after 1% Change
  in After-Tax Interest Rate
  ### Mid-Point Formula

  interface(displayprecision=6):
  schanges := Matrix([seq( [seq( 100*(s(Rf,betaf,Gammaf,rho,mu)-s
  (Rf-1/100,betaf,Gammaf,rho,mu))/((s(Rf,betaf,Gammaf,rho,mu)+s
  (Rf-1/100,betaf,Gammaf,rho,mu))/2) ,rho=rholist[1..8] )],mu=
  mulist[2..8])]):
  schanges := ArrayTools:-Concatenate(2,Vector[column](evalf[2]
  (mulist[2..8])),schanges):
  schanges := ArrayTools:-Concatenate(1,Vector[row]([0,op(rholist
  [1..8])]),schanges):
      'schanges' = evalf(%);
  interface(displayprecision=mydisplayprecision):
```

```
schanges = [[0., 1., 2., 3., 4., 5., 6., 7., 8.],                                        (22)
    [0.010, 5.07042, −0.376677, −0.885073, −0.992096, −1.02230, −1.03092, −1.03219,
    −1.03075],
    [0.020, 3.10618, −0.369990, −0.774884, −0.886073, −0.929742, −0.950550,
    −0.961728, −0.968236],
[0.040, 1.72662, −0.386538, −0.706214, −0.813672, −0.863823, −0.891824,
−0.909325, −0.921145 ],
[0.080, 0.887175, −0.415472, −0.670518, −0.770429, −0.822311, −0.853715,
−0.874641, −0.889534 ],
[0.16, 0.420210, −0.444898, −0.655834, −0.748075, −0.799364, −0.831909,
−0.854360, −0.870770 ],
[0.32, 0.173337, −0.467477, −0.651297, −0.737741, −0.787877, −0.820584,
−0.843597, −0.860667 ],
[0.64, 0.0463261, −0.481930, −0.650408, −0.733188, −0.782385, −0.814986,
−0.838175, −0.855515 ]]
```

```
> #########################
> ### Export Plots
  ### The best quality 2d plots are postscript, the best 3d plots
  are png
  ### figures are converted to pdf or png with epstopdf and
  imagemagick with batch file
> interface(displayprecision=2): # necessary to strip some trailing
  zeros
> MakePlot(mTargetUrateVariesCRRAVaries,'extension'=png); # 3d
  postscript plots buggy in Maple 16 and ugly in earlier versions
> MakePlot(mTargetUrateVariesCRRAVariesAnimation,'extension'=gif);
> MakePlot(mTargetCRRAFixedUrateVaries,'extension'=ps);
> MakePlot(mTargetUrateFixedCRRAVaries,'extension'=ps);
> MakePlot(mTargetCRRAFixedUrateVariesApproximations,'extension'=
  ps);
> MakePlot(mSlopeCRRAFixedUrateVaries,'extension'=ps);
> MakePlot(mSlopeUrateFixedCRRAVaries,'extension'=ps);
> MakePlot(sTargetUrateVariesCRRAVaries,'extension'=png); # 3d
  postscript plots buggy in Maple 16 and ugly in earlier versions
> MakePlot(sTargetUrateVariesCRRAVariesAnimation,'extension'=gif);
> MakePlot(sTargetCRRAFixedUrateVaries,'extension'=ps);
> MakePlot(sTargetUrateFixedCRRAVaries,'extension'=ps);
> MakePlot(sElasticityCRRAFixedUrateVaries,'extension'=ps);
> MakePlot(sElasticityUrateFixedCRRAVaries,'extension'=ps);
> #########################
> ### Export Data to File
  theplace := cat(currentdir(),kernelopts(dirsep),convert(N,
  string),kernelopts(dirsep)):
  thedata := [ 'm'=m(R,beta,Gamma,rho,mu), 's'=s(R,beta,Gamma,rho,
  mu), 'parameters'=params ]:
> fd := fopen(cat(theplace,"ParametersAndFormulas_",convert(N,
  string),".txt"), WRITE):
  fprintf(fd, "%{c\n}a\n", <thedata>): fclose(fd):
> ExportMatrix(cat(theplace,"mvalues_mu_rho_",convert(N,string),".
  m")
      , evalf(mvalues), delimiter="&", format=rectangular, mode=
  ascii):
> ExportMatrix(cat(theplace,"mchanges_mu_rho_",convert(N,string),".
  m")
```

```
      , evalf(mchanges), delimiter="&", format=rectangular, mode=
  ascii):
> ExportMatrix(cat(theplace,"svalues_mu_rho_",convert(N,string),".
  m")
      , evalf(svalues), delimiter="&", format=rectangular, mode=
  ascii):
> ExportMatrix(cat(theplace,"schanges_mu_rho_",convert(N,string),".
  m")
      , evalf(schanges), delimiter="&", format=rectangular, mode=
  ascii):
> interface(displayprecision=mydisplayprecision): # restore
  preferences
```