```
> 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]));
```

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

$$R\,\beta = 0.936 \qquad (1)$$

```
> ### 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]));
```

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

$$R\,\beta = 0.936 \qquad (2)$$

```
> ### 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]));
```

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

$$R\,\beta = 1.01 \qquad (3)$$

```
> ### 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]));
```

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

$$R\,\beta = 0.936 \qquad (4)$$

```
> ### 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]));
```

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

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

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

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

$$params = \left[R = 1.04,\ \beta = 0.975,\ \Gamma = \frac{1.01}{1-\mu}\right] \tag{6}$$

```
> ### 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)**

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

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

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

$$\rho = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] \quad (10)$$

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

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

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

```
> ### 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(%);
```

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

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

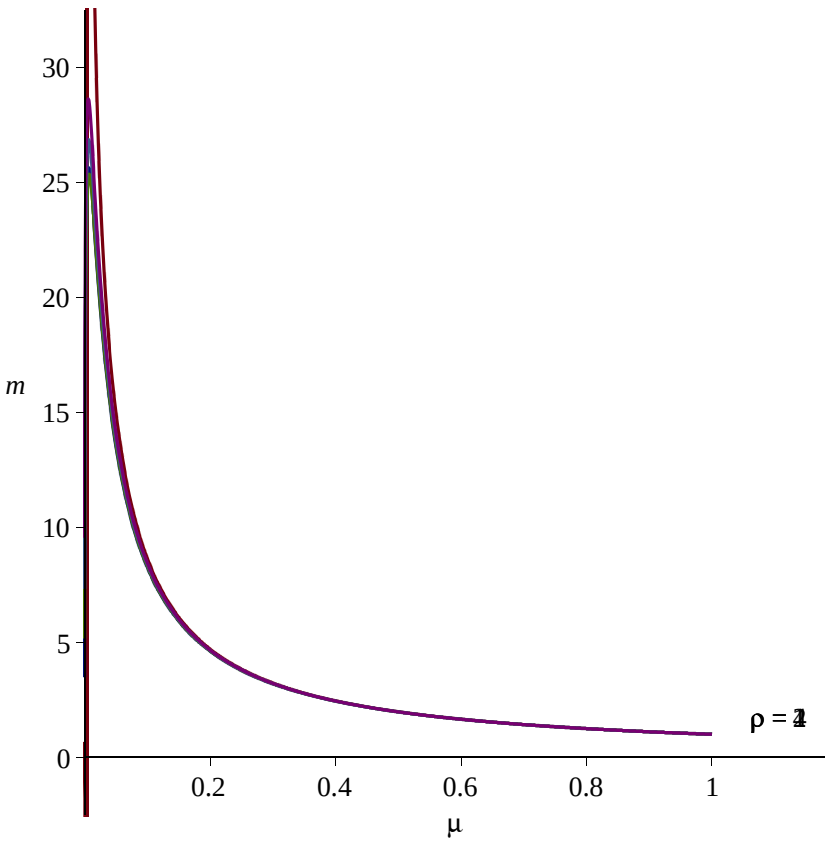
$$\begin{bmatrix} true & true & true & true & true & true & true & true & true & true \\ true & true & true & true & true & true & true & true & true & true \\ true & true & true & true & true & true & true & true & true & true \\ true & true & true & true & true & true & true & true & true & true \\ true & true & true & true & true & true & true & true & true & true \\ true & true & true & true & true & true & true & true & true & true \\ true & true & true & true & true & true & true & true & true & true \end{bmatrix} \quad \textbf{(12)}$$

```
> ### 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):
```

$$m = 1 + \frac{1}{\frac{0.971}{1-\mu} - 1 + \left(1 - 0.962\ 1.01^{\frac{1}{\rho}}\right)\left(1 + \frac{\left(0.990\ 1.01^{\frac{1}{\rho}}\ (1-\mu)\right)^{-\rho} - 1}{\mu}\right)^{\frac{1}{\rho}}} \quad \textbf{(13)}$$

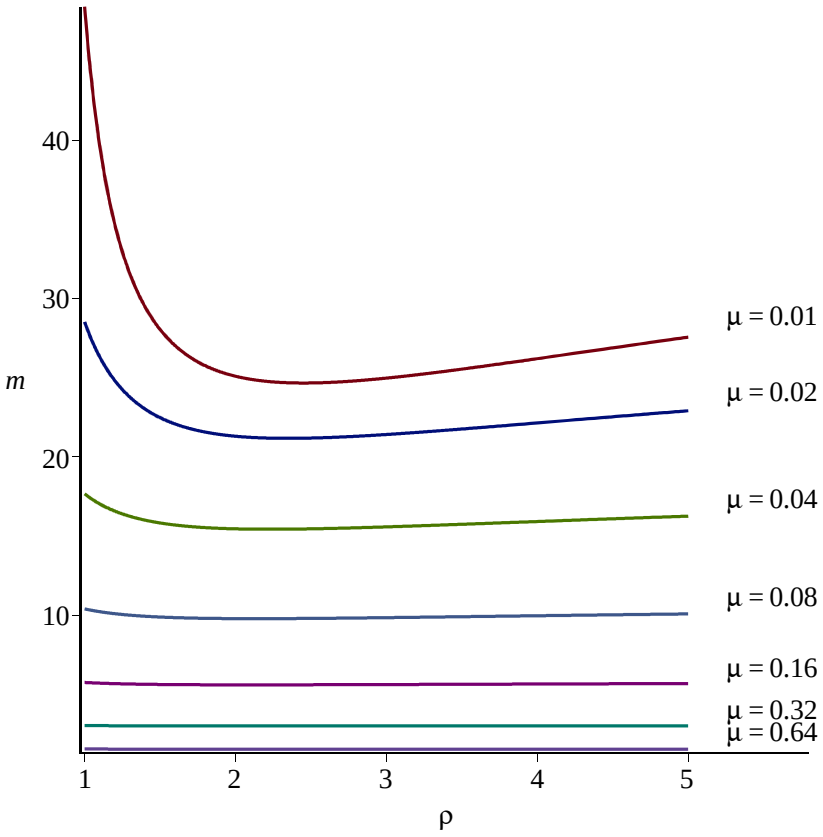
```
> ### Plot of m as rho and mu vary

  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):
```

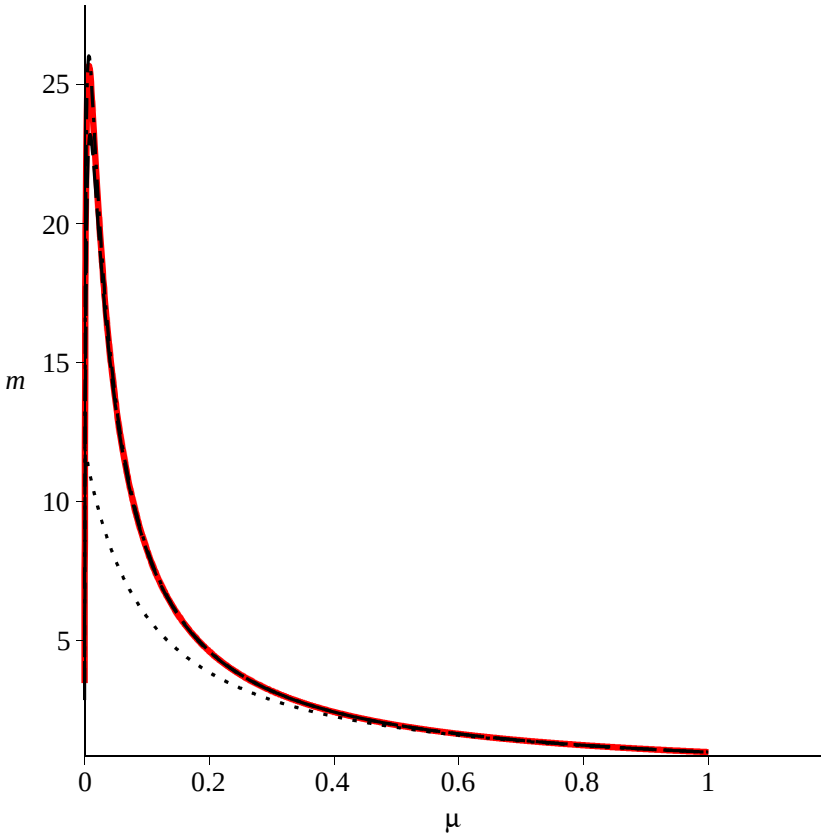
*mvalues* **(14)**

$$=\begin{bmatrix} 0. & 1. & 2. & 3. & 4. & 5. & 6. & 7. & 8. \\ 0.010 & 48.4133 & 25.0904 & 24.9723 & 26.1996 & 27.5598 & 28.8418 & 30.0060 & 31.0535 \\ 0.020 & 28.5253 & 21.3028 & 21.4128 & 22.1420 & 22.9096 & 23.6143 & 24.2417 & 24.7970 \\ 0.040 & 17.6693 & 15.4571 & 15.5811 & 15.9201 & 16.2574 & 16.5582 & 16.8205 & 17.0488 \\ 0.080 & 10.4035 & 9.79166 & 9.85226 & 9.97274 & 10.0876 & 10.1875 & 10.2732 & 10.3466 \\ 0.16 & 5.76027 & 5.61142 & 5.62987 & 5.66238 & 5.69249 & 5.71819 & 5.73982 & 5.75809 \\ 0.32 & 3.04341 & 3.01299 & 3.01590 & 3.02159 & 3.02676 & 3.03104 & 3.03453 & 3.03739 \\ 0.64 & 1.55819 & 1.55384 & 1.55336 & 1.55334 & 1.55339 & 1.55345 & 1.55349 & 1.55353 \end{bmatrix}$$

```
> ### 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 ]
        ) : %;
```

$$mfn = \left[ 1 + \frac{1}{\frac{1}{1-\mu} - 1 + 0.2\sqrt{1 + \frac{\frac{1}{(1-\mu)^2} - 1}{\mu}}}, 1 \right.$$
$$+ \frac{1}{\frac{0.97}{1-\mu} - 1 + 0.07\sqrt{1 + \frac{\frac{1}{(1-\mu)^2} - 1}{\mu}}}, 1$$
$$+ \frac{1}{\frac{0.971}{1-\mu} - 1 + 0.032\sqrt{1 + \frac{\frac{1.01}{(1-\mu)^2} - 1}{\mu}}}, 1$$
$$+ \frac{1}{\frac{0.971}{1-\mu} - 1 + 0.0314\sqrt{1 + \frac{\frac{1.01}{(1-\mu)^2} - 1}{\mu}}}$$
$$\left. + \frac{1}{\frac{0.971}{1-\mu} - 1 + 0.0317\sqrt{1 + \frac{\frac{1.01}{(1-\mu)^2} - 1}{\mu}}} \right]$$

Error, (in mf) numeric exception: division by zero




```
> ##########################
> ### 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])];
```

$$asymptotic\_m\_mu := \left[\frac{101}{2}, \frac{101}{3}, \frac{101}{5}, \frac{101}{9}, \frac{101}{17}, \frac{101}{33}, \frac{101}{65}, \frac{2497}{3225}, \frac{2369}{6425}, \frac{2113}{12825}, \frac{1601}{25625}, \frac{577}{51225}, -\frac{1471}{102425}, -\frac{5567}{204825}, -\frac{13759}{409625}, -\frac{30143}{819225}, -\frac{62911}{1638425}, -\frac{128447}{3276825}, -\frac{19963}{504125}\right] \quad \textbf{(15)}$$

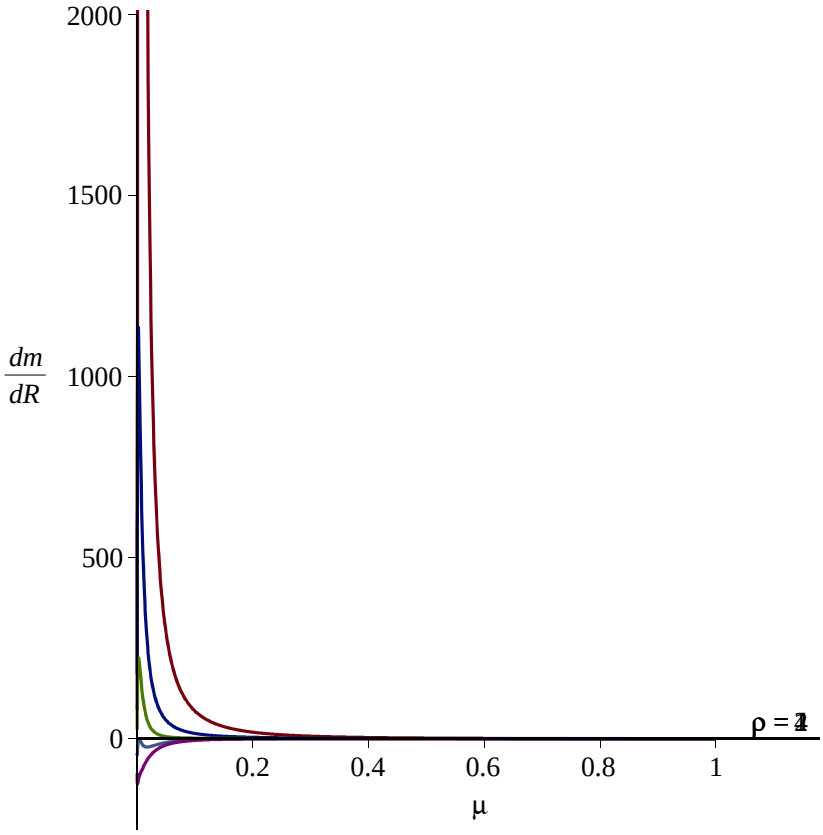
```
> ### Derivative of m with respect to R

  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):
```

$$dm = -\left( -\frac{0.9338}{1-\mu} + \left( -\frac{0.9246\ 1.014^{\frac{1}{\rho}}}{\rho} + 0.9246\ 1.014^{\frac{1}{\rho}} \right) \left( 1 + \frac{\left( 0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu) \right)^{-\rho} - 1}{\mu} \right)^{\frac{1}{\rho}} \right) \Bigg/ \left( \frac{0.9712}{1-\mu} - 1 + \left( 1 - 0.9615\ 1.014^{\frac{1}{\rho}} \right) \left( 1 + \frac{\left( 0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu) \right)^{-\rho} - 1}{\mu} \right)^{\frac{1}{\rho}} \right)^2 \tag{16}$$

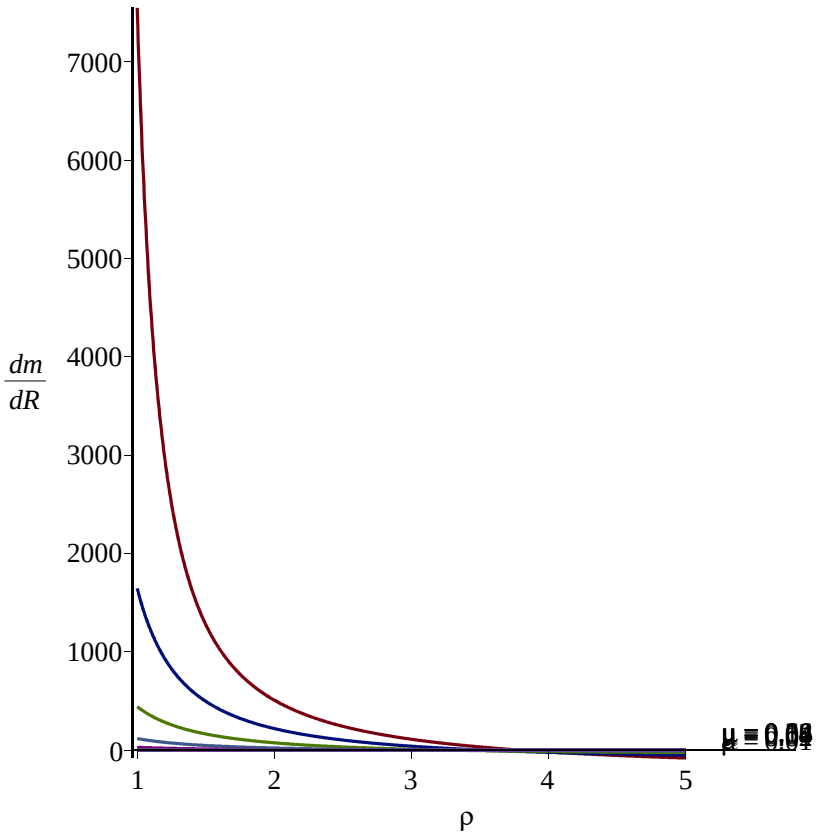
$$- \frac{1}{\rho\,\mu \left( 1 + \frac{\left( 0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu) \right)^{-\rho} - 1}{\mu} \right)} \left( 0.9615 \left( 1 - 0.9615\ 1.014^{\frac{1}{\rho}} \right) \left( 1 + \frac{\left( 0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu) \right)^{-\rho} - 1}{\mu} \right)^{\frac{1}{\rho}} \left( 0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu) \right)^{-\rho} \right) \Bigg/$$

```
> ### 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):
```

$$mchanges = \begin{bmatrix} [0., 1., 2., 3., 4., 5., 6., 7., 8.], \\ [0.010, 86.4847, 15.3037, 2.60074, -1.78702, -3.70419, -4.63079, -5.08973, \\ -5.30450], \\ [0.020, 44.3915, 8.52562, 1.10453, -1.52148, -2.65456, -3.18177, -3.42398, \\ -3.51942], \end{bmatrix} \quad \textbf{(17)}$$

```
    [0.040, 22.1835, 4.29426, 0.350269, −1.04013, −1.62180, −1.87652, −1.97972,
    −2.00669],
    [0.080, 10.7490, 2.11244, 0.174259, −0.498246, −0.770259, −0.882286, −0.921555,
    −0.925167],
    [0.16, 4.94237, 1.07593, 0.204458, −0.0957231, −0.216184, −0.265626, −0.283230,
    −0.285514],
    [0.32, 2.01361, 0.553749, 0.217964, 0.0977127, 0.0453213, 0.0198715, 0.00667080,
    −0.000449398],
    [0.64, 0.537941, 0.210422, 0.125157, 0.0885455, 0.0685260, 0.0559248, 0.0472528,
    0.0409134]]
> ##########################
> ### 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):
```

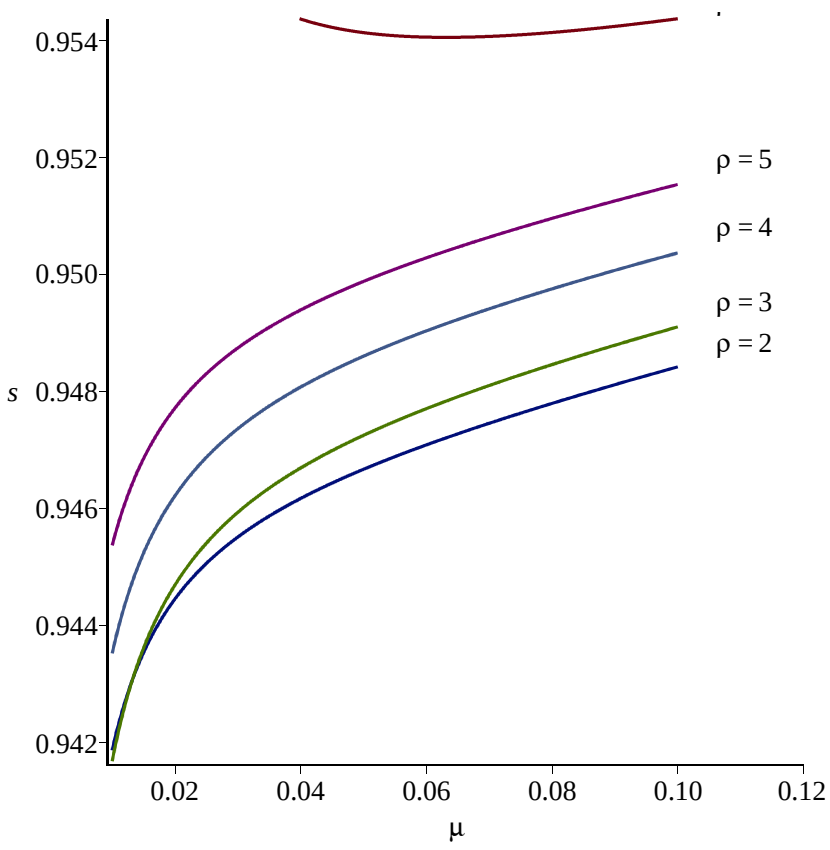
$$s = \frac{1}{1 + 1.030\left(1 - 0.9615\,1.014^{\frac{1}{\rho}}\right)(1-\mu)\left(\frac{\left(0.9901\,1.014^{\frac{1}{\rho}}(1-\mu)\right)^{-\rho} - 1 + \mu}{\mu}\right)^{\frac{1}{\rho}}} \qquad \textbf{(18)}$$

```
> ### 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;
```

*plot_s_rho_mu* **(19)**

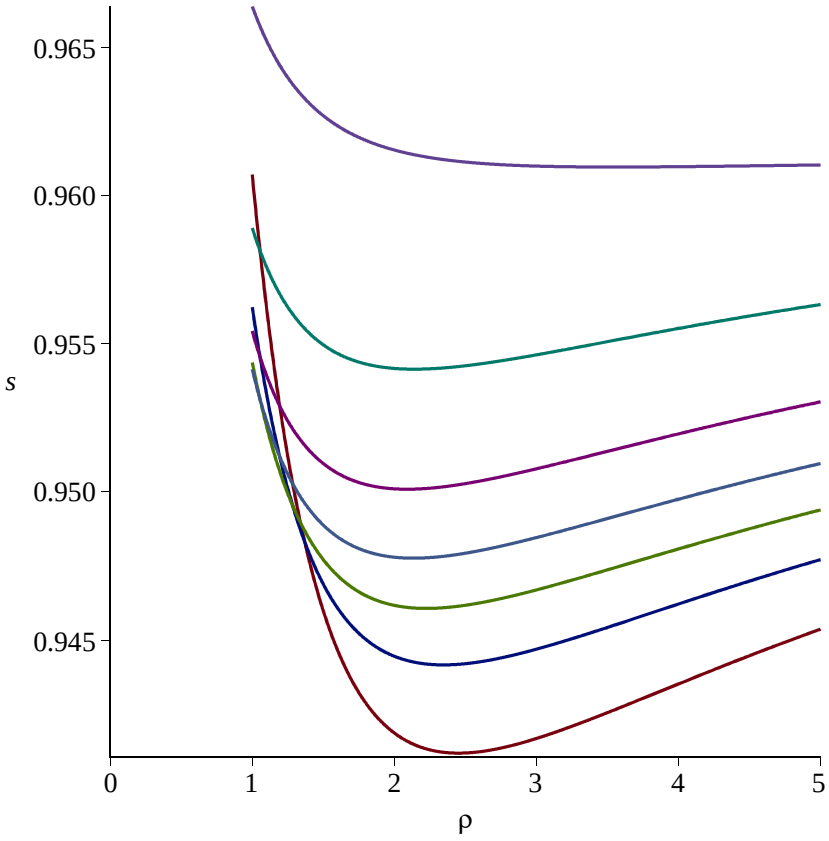
```
> ### Animated plot of m as rho and mu vary

  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, 0.960701, 0.941866, 0.941681, 0.943521, 0.945370, 0.946952, 0.948271, 0.949374
],
[0.020, 0.956233, 0.944455, 0.944694, 0.946218, 0.947718, 0.949008, 0.950094, 0.951010
],
[0.040, 0.954366, 0.946172, 0.946692, 0.948075, 0.949393, 0.950524, 0.951476, 0.952282
],
[0.080, 0.954136, 0.947796, 0.948459, 0.949753, 0.950958, 0.951985, 0.952849, 0.953578
],
[0.16, 0.955427, 0.950103, 0.950778, 0.951957, 0.953037, 0.953950, 0.954712, 0.955351 ],
[0.32, 0.958901, 0.954164, 0.954622, 0.955513, 0.956320, 0.956987, 0.957529, 0.957972 ],
[0.64, 0.966375, 0.961532, 0.960990, 0.960970, 0.961029, 0.961092, 0.961145, 0.961188 ] ]
```

```
> ### 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):
```

$$es = -\left(1.040\left(1.030\left(-\frac{0.9246\ 1.014^{\frac{1}{\rho}}}{\rho}+0.9246\ 1.014^{\frac{1}{\rho}}\right)(1-\mu)\left(\frac{\left(0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu)\right)^{-\rho}-1+\mu}{\mu}\right)^{\frac{1}{\rho}}+0.9901\left(1-0.9615\ 1.014^{\frac{1}{\rho}}\right)(1-\mu)\left(\frac{\left(0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu)\right)^{-\rho}-1+\mu}{\mu}\right)^{\frac{1}{\rho}}\right.\right. \tag{21}$$

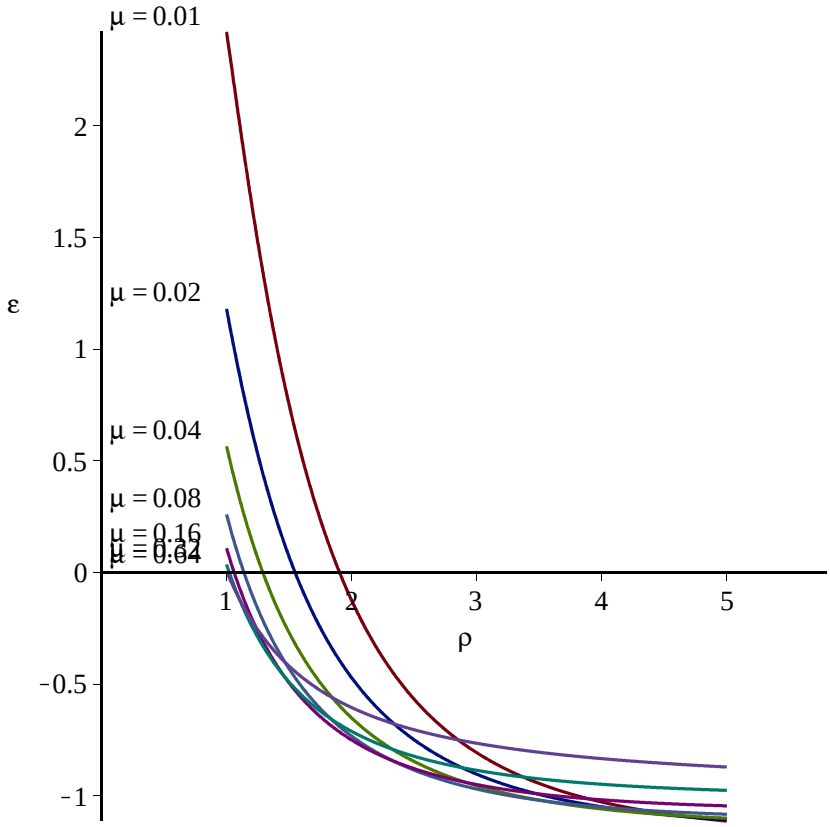
$$-\frac{1}{\rho\left(\left(0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu)\right)^{-\rho}-1+\mu\right)}\left(0.9901\left(1-0.9615\ 1.014^{\frac{1}{\rho}}\right)(1-\mu)\left(\frac{\left(0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu)\right)^{-\rho}-1+\mu}{\mu}\right)^{\frac{1}{\rho}}\left(0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu)\right)^{-\rho}\right)\right)$$

$$\Big/$$

$$\left(1+1.030\left(1-0.9615\ 1.014^{\frac{1}{\rho}}\right)(1-\mu)\left(\frac{\left(0.9901\ 1.014^{\frac{1}{\rho}}\ (1-\mu)\right)^{-\rho}-1+\mu}{\mu}\right)^{\frac{1}{\rho}}\right)$$

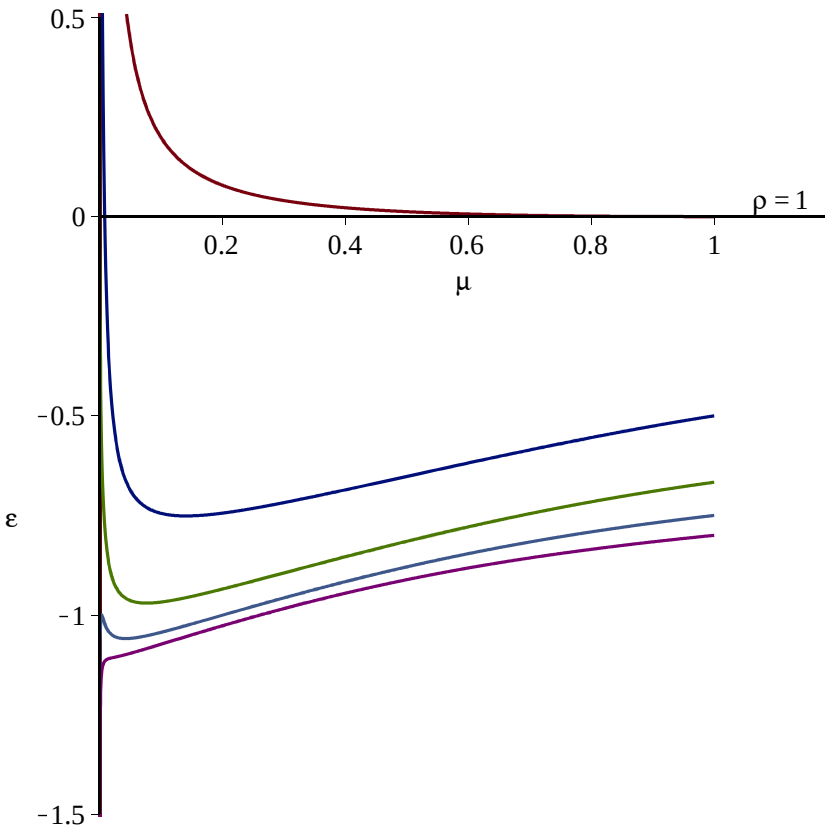
```
> ### Set position of the plot labels, tweaked for stated parameter
  values

  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):
```

$$schanges = \begin{bmatrix} [0., 1., 2., 3., 4., 5., 6., 7., 8.], \\ [0.010, 2.30025, -0.275915, -0.856206, -1.03644, -1.10302, -1.12861, -1.13715, \\ -1.13797], \end{bmatrix} \quad (22)$$

```
    [0.020, 1.12842, −0.526604, −0.911760, −1.03758, −1.08568, −1.10458, −1.11092,
    −1.11141],
    [0.040, 0.541950, −0.662175, −0.942117, −1.03551, −1.07157, −1.08560, −1.09001,
    −1.08990],
    [0.080, 0.249170, −0.723050, −0.946480, −1.02156, −1.05058, −1.06174, −1.06506,
    −1.06466],
    [0.16, 0.104076, −0.731333, −0.921969, −0.986702, −1.01219, −1.02239, −1.02584,
    −1.02608],
    [0.32, 0.0342380, −0.689955, −0.857887, −0.917815, −0.943815, −0.956398,
    −0.962905, −0.966404],
    [0.64, 0.00483616, −0.585131, −0.739611, −0.805967, −0.842238, −0.865058,
    −0.880757, −0.892230]]
> ##########################
> ### 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
```