TPSS

The original Tpss code is copyrighted, I belive that suche things hinder the advance of science, so here is (I think) a nice way to generate code that calculates the TPSS functional that can be used by anyone following the GPL 2 or higher license.

"Progress in science through free software" (isn't it Joost?;)

Fawzi Mohamed

Reference: Perdew,Tao, Staroverov, Scuseria, J.Chem Phys vol 120, p 6898 (2004), folowup article of Perdew

```
> restart;
```

```
> sost:=eqs ->
  subs(seq(eqs[nops(eqs)-i],i=1..(nops(eqs)-1)),rhs(eqs[nops(eqs)])):
```

```
> unk:=eqs -> indets(sost(eqs),symbol):
```

```
> loc:=eqs -> indets(eqs,symbol) minus unk(eqs):
```

```
> e:='e': m:='m': h_bar:='h_bar': a_0:='a_0':
```

```
> e:=1: m:=1: h_bar:=1: a_0:=h_bar^2 /(m*e^2):
```

```
> indice:=proc(el,l) local i,ii,elAtt,el_s;
   i:=-1; ii:=0; el_s:=convert(el,string);
   for elAtt in l do
    ii:=ii+1;
    if evalb(el_s=convert(elAtt,string)) then
      i:=ii;
    end if;
   end do;
   i;
  end proc:
```

```
> indiceDef:=proc(el,l) local i,ii,elAtt;
   i:=-1; ii:=0;
   for elAtt in l do
    ii:=ii+1;
    if evalb(el=lhs(elAtt)) then
      i:=ii;
    end if;
   end do;
   i;
  end proc:
```

```
> definizioni:= eqs -> map(eq -> if type(eq,equation) then lhs(eq); else
  0; end if ,eqs):
```

```
> sameNameSameDef:=proc(eqs1,eqs2) local commonDef,res,d;
   commonDef:=convert(definizioni(eqs1),set)intersect
  convert(definizioni(eqs2),set);
   res:=true;
   for d in commonDef do
    if not evalb(subs(eqs1,d)=subs(eqs2,d)) then;
     print("def different for "||d);
     res:=false;
    end if;
   end do;
   res;
  end proc:
```

```
> # check same name -> same def apart from eqs at the indexes returned
  by the function eqs_to_rm
  checkCompatible:=proc (eqss,eqs_to_rm) local
  i,j,im_indx,eqd1,eqd2,res,ii;
    res:=true;
    for i from 1 to nops(eqss)-1 do
      im_indx:=eqs_to_rm(eqss[i]);

  #print("removed",map(lhs,[eqss[i][im_indx[ii]]$ii=1..nops(im_indx)]));
      eqd1:=subsop('im_indx[ii]=NULL'$ii=1..nops(im_indx),eqss[i]):
      for j from i+1 to nops(eqss) do
        #print("doing (",i,j,")");
        im_indx:=eqs_to_rm(deriv_lda[i1,j1][ideriv]);

  #print("removed",map(lhs,[eqss[j][im_indx[ii]]$ii=1..nops(im_indx)]));
        eqd2:=subsop('im_indx[ii]=NULL'$ii=1..nops(im_indx),eqss[j]):
        res:=sameNameSameDef(eqd1,eqd2) and res;
      end do;
    end do;
    res;
  end proc:
```

```
> getDef:=proc(symb,eqs) local eq;
   for eq in eqs do
    if(lhs(eq)=symb) then
     return eq;
    end if;
   end do;
   0;
  end proc:
```

```
> eqUses:=(eq1,eq2)->evalb(lhs(eq2) in indets(rhs(eq1),symbol)):
```

```
> enforceDependencies:=proc(eqs) local dep,eq1,eq2,i,j,ii,eqns;
   dep:=true;
   eqns:=eqs;
   ii:=0;
   i:=1;
   while (i<=(nops(eqs)-1) and ii<10000) do
    dep:=false;
    j:=i+1;
    while (j<=nops(eqs) and ii<10000) do
     if eqUses(eqns[i],eqns[j]) then
      ii:=ii+1;
      eqns:=subsop(i=NULL,j=(eqns[j],eqns[i]),eqns);
      dep:=true;
     else
      j:=j+1;
     end if;
    end do;
    if not dep then i:=i+1; end if;
   end do;
   eqns;
  end proc:
```

```
> combineDefs:=proc(ord) local def,defs,allDefs;
   allDefs:=[]:
   for defs in ord do
    for def in defs do
     if not def in allDefs then
      allDefs:=[op(allDefs),def];
     end if;
    end do;
   end do;
   allDefs;
  end proc:
```

```
> combineEqs:=proc(allDefs,eqss,ord) local
  def,eqs,eqsDeriv,found,d,i,ii;
   eqsDeriv:=[];
   for def in allDefs do
    found:=false;
    for ii from 1 to nops(eqss) do
     d:=ord[ii];
     i:=indice(def,d);
     if i>0 then
      eqs:=eqss[ii];
      if not (lhs(eqs[i])=def) then print("errore eq",def); end if;
      eqsDeriv:=[op(eqsDeriv),eqs[i]];
      found:=true;
      break;
     end if;
    end do;
    if not found then print("error unknown def",def); end if;
   end do;
   eqsDeriv;
  end proc:
```

```
> sostConst:=proc(eqs) local sAtt,sToDo,result;
   sToDo:=[];
   result:=[];
   for sAtt in eqs do
    sAtt:=subs(op(sToDo),sAtt);
    if type(rhs(sAtt),numeric) then sToDo:=[op(sToDo),sAtt]; end if;
    if rhs(sAtt)<>0 then result:=[op(result),sAtt]; end if;
   end do;
   result;
  end proc:
```

```
> with(CodeGeneration);
Warning, the protected name Matlab has been redefined and unprotected

     [C, Fortran, IntermediateCode, Java, LanguageDefinition, Matlab, Names, Translate, VisualBasic]
```

exchange

exhange energy (LDA)

```
> eqx1:=ex_lda=rho*ex_unif*Fx;
```

$$eqx1 := ex\_lda = \rho\ ex\_unif\ Fx$$

Uniform gas exchange:

```
> eqx2:=ex_unif=-3/(4*Pi)*(3*Pi^2*rho)^(1/3);
```

$$eqx2 := ex\_unif = -\frac{3\ 3^{(1/3)}\ (\pi^2\ \rho)^{(1/3)}}{4\ \pi}$$

The enhancement factor Fx is function of just p an z;

```
> eqx3:=p=norm_drho^2/(4*(3*Pi^2)^(2/3)*rho^(8/3));
  eqx4:=s=(3/Pi)^(2/3)/6*norm_drho/rho^(4/3);
  eqx5:=z=tau_w/tau;
  eqx6:=tau_w=norm_drho^2/(8*rho);
```

$$eqx3 := p = \frac{norm\_drho^2\ 3^{(1/3)}}{12\ (\pi^2)^{(2/3)}\ \rho^{(8/3)}}$$

$$eqx4 := s = \frac{3^{(2/3)} \left(\frac{1}{\pi}\right)^{(2/3)} norm\_drho}{6\ \rho^{(4/3)}}$$

$$eqx5 := z = \frac{tau\_w}{\tau}$$

$$eqx6 := tau\_w = \frac{norm\_drho^2}{8\ \rho}$$

```
> evalb(simplify(subs(eqx3,eqx4,s^2=p),symbolic));
```

$$true$$

```
> eqx7:=tildeq_b=(9/20)*(alpha-1)/(1+b*alpha*(alpha-1))^(1/2)+2*p/3;
  eqx8:=alpha=(5*p/3)*(z ^(-1)-1); # =(tau-tau_w)/tau_unif
>
```

$$eqx7 := tildeq\_b = \frac{9\ (\alpha - 1)}{20\ \sqrt{1 + b\ \alpha\ (\alpha - 1)}} + \frac{2}{3} p$$

$$eqx8 := \alpha = \frac{5}{3} p \left(\frac{1}{z} - 1\right)$$

Fx can be written as

```
> eqx9:=Fx=1+kappa-kappa/(1+x/kappa);
  eqk1:=kappa=0.804;
  eqk2:=mu=0.21951;
```

$$eqx9 := Fx = 1 + \kappa - \frac{\kappa}{1 + \frac{x}{\kappa}}$$

$$eqk1 := \kappa = 0.804$$

$$eqk2 := \mu = 0.21951$$

and x

```
> eqx10:=x=((10/81+c*z^2/(1+z^2)^2)*p+146/2025*tildeq_b^2-73/405*tild
  eq_b*sqrt(1/2*(3/5*z)^2+1/2*p^2)+1/kappa*(10/81)^2*p^2+
  2*sqrt(e_var)*10/81*(3/5*z)^2+e_var*mu*p^3)/(1+sqrt(e_var)*p)^2;
```

$$eqx10 := x = \frac{1}{(1 + \sqrt{e\_var}\, p)^2} \left(\left(\frac{10}{81} + \frac{c\, z^2}{(1 + z^2)^2} \right) p + \frac{146}{2025} tildeq\_b^2 - \frac{73}{4050} tildeq\_b \sqrt{18 z^2 + 50 p^2} + \frac{100\, p^2}{6561\, \kappa} + \frac{4}{45} \sqrt{e\_var}\, z^2 + e\_var\, \mu\, p^3 \right)$$

```
> eqk3:=b=0.4;
  eqk4:=c=1.59096;
  eqk5:=e_var=1.537;
```

$$eqk3 := b = 0.4$$

$$eqk4 := c = 1.59096$$

$$eqk5 := e\_var = 1.537$$

```
> eqs_ex_lda := [eqk1,eqk2,eqk3,eqk4,eqk5,eqx3, eqx6, eqx5, eqx8,
  eqx7, eqx10, eqx9,eqx2,eqx1];
```

$$eqs\_ex\_lda := \left[\kappa = 0.804,\ \mu = 0.21951,\ b = 0.4,\ c = 1.59096,\ e\_var = 1.537,\ p = \frac{norm\_drho^2\, 3^{(1/3)}}{12\, (\pi^2)^{(2/3)}\, \rho^{(8/3)}}, \right.$$

$$tau\_w = \frac{norm\_drho^2}{8\, \rho},\ z = \frac{tau\_w}{\tau},\ \alpha = \frac{5}{3} p \left(\frac{1}{z} - 1 \right),\ tildeq\_b = \frac{9\, (\alpha - 1)}{20 \sqrt{1 + b\, \alpha\, (\alpha - 1)}} + \frac{2}{3} p,\ x =$$

$$\frac{1}{(1 + \sqrt{e\_var}\, p)^2} \left(\left(\frac{10}{81} + \frac{c\, z^2}{(1 + z^2)^2} \right) p + \frac{146}{2025} tildeq\_b^2 - \frac{73}{4050} tildeq\_b \sqrt{18 z^2 + 50 p^2} \right.$$

$$\left. + \frac{100\, p^2}{6561\, \kappa} + \frac{4}{45} \sqrt{e\_var}\, z^2 + e\_var\, \mu\, p^3 \right),\ Fx = 1 + \kappa - \frac{\kappa}{1 + \frac{x}{\kappa}},\ ex\_unif = -\frac{3\, 3^{(1/3)}\, (\pi^2\, \rho)^{(1/3)}}{4\, \pi},$$

$$\mathrm{ex\_lda} = \rho\ \mathrm{ex\_unif}\,\mathrm{Fx}$$

```
> unk(eqs_ex_lda);
```

$$\{\mathrm{norm\_drho}, \rho, \tau, \pi\}$$

```
> loc(eqs_ex_lda);
```

$$\{\mathrm{ex\_lda}, \mathrm{e\_var}, \mathrm{ex\_unif}, \mathrm{tau\_w}, \kappa, \alpha, \mathrm{Fx}, z, p, \mu, x, \mathrm{tildeq\_b}, b, c\}$$

correlation

```
> eqc1:=ec=rho*epsilon_cRevPKZB*(1+d*epsilon_cRevPKZB*(tau_w/tau)^3);
```

$$\mathrm{eqc1} := \mathrm{ec} = \rho\ \mathrm{epsilon\_cRevPKZB}\left(1 + \frac{d\ \mathrm{epsilon\_cRevPKZB}\ \mathrm{tau\_w}^3}{\tau^3}\right)$$

```
> eqc2_1:=ma=max(epsilon_cGGA_1_0,epsilon_cGGA);
  eqc2_2:=mb=max(epsilon_cGGA_0_1,epsilon_cGGA);
  eqc2:=epsilon_cRevPKZB=epsilon_cGGA*(1+C_chi_eps*(tau_w/tau)^2)-(1+
  C_chi_eps)*(tau_w/tau)^2*(rhoa/rho*ma+rhob/rho*mb);
```

$$\mathrm{eqc2\_1} := \mathrm{ma} = \max(\mathrm{epsilon\_cGGA\_1\_0}, \mathrm{epsilon\_cGGA})$$

$$\mathrm{eqc2\_2} := \mathrm{mb} = \max(\mathrm{epsilon\_cGGA\_0\_1}, \mathrm{epsilon\_cGGA})$$

$$\mathrm{eqc2} := \mathrm{epsilon\_cRevPKZB} = \mathrm{epsilon\_cGGA}\left(1 + \frac{\mathrm{C\_chi\_eps}\ \mathrm{tau\_w}^2}{\tau^2}\right) - \frac{(1 + \mathrm{C\_chi\_eps})\ \mathrm{tau\_w}^2\left(\frac{\mathrm{rhoa}\ \mathrm{ma}}{\rho} + \frac{\mathrm{rhob}\ \mathrm{mb}}{\rho}\right)}{\tau^2}$$

```
> eqc3:=chi=(rhoa-rhob)/rho;
  eqc4:=eps=norm_dchi/(2*(3*Pi^2*rho)^(1/3));
```

$$\mathrm{eqc3} := \chi = \frac{\mathrm{rhoa} - \mathrm{rhob}}{\rho}$$

$$\mathrm{eqc4} := \mathrm{eps} = \frac{\mathrm{norm\_dchi}\ 3^{(2/3)}}{6\ (\pi^2\ \rho)^{(1/3)}}$$

```
> eqc5:=C_chi=0.53+0.87*chi^2+0.5*chi^4+2.26*chi^6;
  eqc6:=C_chi_eps=C_chi/(1+eps^2*((1+chi)^(-4/3)+(1-chi)^(-4/3))/2)^4
  ;
```

$$\mathrm{eqc5} := \mathrm{C\_chi} = 0.53 + 0.87\ \chi^2 + 0.5\ \chi^4 + 2.26\ \chi^6$$

$$eqc6 := C\_chi\_eps = \frac{C\_chi}{\left(1+\frac{1}{2}\,eps^{2}\left(\frac{1}{(1+\chi)^{(4/3)}}+\frac{1}{(1-\chi)^{(4/3)}}\right)\right)^{4}}$$

```
> eqc7:=rs=(3/(4*Pi*rho))^(1/3);
```

$$eqc7 := rs = \frac{1}{4}\,3^{(1/3)}\,4^{(2/3)}\left(\frac{1}{\pi\,\rho}\right)^{(1/3)}$$

```
> eqc8:=d=2.8;
```

$$eqc8 := d = 2.8$$

```
> eqc9:=norm_dchi=2*sqrt((norm_drhoa*rhob)^2+(norm_drhob*rhoa)^2
    -(norm_drho^2-norm_drhoa^2-norm_drhob^2)*rhoa*rhob)/rho^2;
```

$$eqc9 := norm\_dchi = \frac{1}{\rho^{2}}(2\,\mathrm{sqrt}(norm\_drhoa^{2}\,rhob^{2} + norm\_drhob^{2}\,rhoa^{2} - rhoa\,rhob\,norm\_drho^{2} + rhoa\,rhob\,norm\_drhoa^{2} + rhoa\,rhob\,norm\_drhob^{2}))$$

```
> eqs_c1:=[eqc8,eqc3,eqc9,eqc4,eqc5,eqx6,eqc6,eqc2_1,eqc2_2,eqc2,eqc1
  ];
```

$$eqs\_c1 := \Big[\, d = 2.8,\ \chi = \frac{rhoa - rhob}{\rho},\ norm\_dchi = \frac{1}{\rho^{2}}(2\,\mathrm{sqrt}(norm\_drhoa^{2}\,rhob^{2} + norm\_drhob^{2}\,rhoa^{2} - rhoa\,rhob\,norm\_drho^{2} + rhoa\,rhob\,norm\_drhoa^{2} + rhoa\,rhob\,norm\_drhob^{2})),\ eps = \frac{norm\_dchi\ 3^{(2/3)}}{6\,(\pi^{2}\,\rho)^{(1/3)}},$$

$$C\_chi = 0.53 + 0.87\,\chi^{2} + 0.5\,\chi^{4} + 2.26\,\chi^{6},\ tau\_w = \frac{norm\_drho^{2}}{8\,\rho},$$

$$C\_chi\_eps = \frac{C\_chi}{\left(1+\frac{1}{2}\,eps^{2}\left(\frac{1}{(1+\chi)^{(4/3)}}+\frac{1}{(1-\chi)^{(4/3)}}\right)\right)^{4}},$$

$$ma = \max(epsilon\_cGGA\_1\_0,\ epsilon\_cGGA),$$

$$mb = \max(epsilon\_cGGA\_0\_1,\ epsilon\_cGGA),\ epsilon\_cRevPKZB = epsilon\_cGGA\ \Big(\ 1$$

$$+\frac{\mathrm{C\_chi\_eps}\ \mathrm{tau\_w}^{2}}{\tau^{2}}\Bigg) - \frac{(1+\mathrm{C\_chi\_eps})\ \mathrm{tau\_w}^{2}\left(\frac{\mathrm{rhoa\ ma}}{\rho}+\frac{\mathrm{rhob\ mb}}{\rho}\right)}{\tau^{2}},$$

$$\mathrm{ec} = \rho\ \mathrm{epsilon\_cRevPKZB}\left(1+\frac{\mathrm{d\ epsilon\_cRevPKZB}\ \mathrm{tau\_w}^{3}}{\tau^{3}}\right)\Bigg]$$

```
> unk(eqs_c1);
```

{norm_drho, ρ, τ, epsilon_cGGA_0_1, π, epsilon_cGGA_1_0, rhoa, rhob, epsilon_cGGA, norm_drhoa, norm_drhob}

PBE (alias epsilon_cGGA) from Perdew, Burke, Ernzehof, PRL, vol 77, p 3865 (1996) It has some corrections and discussions: to do, check the value of the constants to use!

```
> eqpbe1:=t=norm_drho/(2*phi*k_s*rho);
```

$$\mathrm{eqpbe1} := t = \frac{\mathrm{norm\_drho}}{2\ \phi\ \mathrm{k\_s}\ \rho}$$

```
> eqpbe2:=phi=((1+chi)^(2/3)+(1-chi)^(2/3))/2;
```

$$\mathrm{eqpbe2} := \phi = \frac{1}{2}(1+\chi)^{(2/3)} + \frac{1}{2}(1-\chi)^{(2/3)}$$

```
> eqpbe3:=k_s=sqrt(4*k_f/(Pi*a_0));
  #eqpbe4:=a_0=h_bar^2 /(m*e^2);
```

$$\mathrm{eqpbe3} := \mathrm{k\_s} = 2\sqrt{\frac{\mathrm{k\_f}}{\pi}}$$

```
> eqpbe5:=H=(e^2/a_0)*gamma_var*phi^3*ln(1+beta/gamma_var*t^2*(1+A*t^
  2)/(1+A*t^2+A^2*t^4));
```

$$\mathrm{eqpbe5} := H = \mathrm{gamma\_var}\ \phi^{3}\ \ln\left(1+\frac{\beta\ t^{2}\ (1+A\ t^{2})}{\mathrm{gamma\_var}\ (1+A\ t^{2}+A^{2}\ t^{4})}\right)$$

```
> eqpbe6:=A=beta/gamma_var*(exp(-epsilon_c_unif/(gamma_var*phi^3*e^2/
  a_0))-1)^(-1);
```

$$\mathrm{eqpbe6} := A = \frac{\beta}{\mathrm{gamma\_var}\left(e^{\left(-\frac{\mathrm{epsilon\_c\_unif}}{\mathrm{gamma\_var}\ \phi^{3}}\right)} - 1\right)}$$

```
> eqpbe7:=epsilon_cGGA=epsilon_c_unif+H;
```

$$\mathrm{eqpbe7} := \mathrm{epsilon\_cGGA} = \mathrm{epsilon\_c\_unif} + H$$

```
> eqpbe8:=beta=0.066725;
```

```
eqpbe9:=gamma_var=(1-ln(2))/Pi^2;evalf(rhs(eqpbe9));
```

$$\mathrm{eqpbe8} := \beta = 0.066725$$

$$\mathrm{eqpbe9} := \mathrm{gamma\_var} = \frac{1 - \ln(2)}{\pi^{2}}$$

$$0.03109069086$$

```
> eqpbe10:=k_f=(3*Pi^2*rho)^(1/3);
```

$$\mathrm{eqpbe10} := \mathrm{k\_f} = 3^{(1/3)} \, (\pi^{2} \rho)^{(1/3)}$$

```
> eqs_pbec1 := [eqpbe8,eqpbe9,eqc3, eqpbe2, eqpbe10, eqpbe3, eqpbe1,
  eqpbe6, eqpbe5,eqpbe7];
```

$$\mathrm{eqs\_pbec1} := \left[\beta = 0.066725, \mathrm{gamma\_var} = \frac{1 - \ln(2)}{\pi^{2}}, \chi = \frac{\mathrm{rhoa} - \mathrm{rhob}}{\rho}, \phi = \frac{1}{2}(1 + \chi)^{(2/3)} + \frac{1}{2}(1 - \chi)^{(2/3)}, \right.$$

$$\mathrm{k\_f} = 3^{(1/3)} \, (\pi^{2} \rho)^{(1/3)}, \mathrm{k\_s} = 2\sqrt{\frac{\mathrm{k\_f}}{\pi}}, t = \frac{\mathrm{norm\_drho}}{2\, \phi \, \mathrm{k\_s}\, \rho},$$

$$A = \frac{\beta}{\mathrm{gamma\_var}\left(e^{\left(-\frac{\mathrm{epsilon\_c\_unif}}{\mathrm{gamma\_var}\, \phi^{3}}\right)} - 1 \right)},$$

$$H = \mathrm{gamma\_var}\, \phi^{3} \ln\left(1 + \frac{\beta\, t^{2} (1 + A\, t^{2})}{\mathrm{gamma\_var}\, (1 + A\, t^{2} + A^{2}\, t^{4})} \right),$$

$$\left. \mathrm{epsilon\_cGGA} = \mathrm{epsilon\_c\_unif} + H \right]$$

```
> unk(eqs_pbec1);
```

$$\{\mathrm{norm\_drho}, \rho, \mathrm{epsilon\_c\_unif}, \pi, \mathrm{rhoa}, \mathrm{rhob}\}$$

Uniform gas correlation from Perdew,Wang; PRB vol 45, p 13244, 1992

```
> equc1:=epsilon_c_unif=e_c_u_0+alpha_c*f/f_ii_0*(1-chi^4)+(e_c_u_1-e
  _c_u_0)*f*chi^4;
```

$$\mathrm{equc1} := \mathrm{epsilon\_c\_unif} = \mathrm{e\_c\_u\_0} + \frac{\mathrm{alpha\_c}\, f\, (1 - \chi^{4})}{\mathrm{f\_ii\_0}} + (\mathrm{e\_c\_u\_1} - \mathrm{e\_c\_u\_0})\, f \chi^{4}$$

```
> equc2:=f=((1+chi)^(4/3)+(1-chi)^(4/3)-2)/(2^(4/3)-2);
```

$$\mathrm{equc2} := f = \frac{(1+\chi)^{(4/3)} + (1-\chi)^{(4/3)} - 2}{2\;2^{(1/3)} - 2}$$

```
> equc3:=f_ii_0=subs(chi=0,diff(subs(equc2,f),chi,chi));
  evalf(rhs(equc3));
```

$$\mathrm{equc3} := \mathrm{f\_ii\_0} = \frac{8}{9\left(2\;2^{(1/3)} - 2\right)}$$

$$1.709920933$$

```
> G_uc:=-2*A*(1+alpha_1*rs)*ln(1+1/(2*A*(beta_1*rs^(1/2)+beta_2*rs+be
  ta_3*rs^(3/2)+beta_4*rs^(p+1))));
```

$$\mathrm{G\_uc} := -2\,A\,(1 + \mathrm{alpha\_1}\ \mathrm{rs})\ \ln\left(1 + \frac{1}{2\,A\left(\mathrm{beta\_1}\,\sqrt{\mathrm{rs}} + \mathrm{beta\_2}\ \mathrm{rs} + \mathrm{beta\_3}\ \mathrm{rs}^{(3/2)} + \mathrm{beta\_4}\ \mathrm{rs}^{(p\,+\,1)}\right)}\right)$$

```
> equc4:={p=1.0,A=0.031091,alpha_1=0.21370,beta_1=7.5957,beta_2=3.587
  6,beta_3=1.6382,
    beta_4=0.49294};
  equc5:=e_c_u_0=subs(equc4,G_uc);
```

$$\mathrm{equc4} := \{p = 1.0, A = 0.031091, \mathrm{alpha\_1} = 0.21370, \mathrm{beta\_1} = 7.5957, \mathrm{beta\_2} = 3.5876, \mathrm{beta\_3} = 1.6382, \mathrm{beta\_4} = 0.49294\}$$

$$\mathrm{equc5} := \mathrm{e\_c\_u\_0} = -0.062182\,(1 + 0.21370\ \mathrm{rs})\ \ln\left(1 + \frac{16.08182432}{7.5957\,\sqrt{\mathrm{rs}} + 3.5876\ \mathrm{rs} + 1.6382\ \mathrm{rs}^{(3/2)} + 0.49294\ \mathrm{rs}^{2.0}}\right)$$

```
> equc6:={p=1.0,A=0.015545,alpha_1=0.20548,beta_1=14.1189,beta_2=6.19
  77,beta_3=3.3662,
    beta_4=0.62517};
  equc7:=e_c_u_1=subs(equc6,G_uc);
```

$$\mathrm{equc6} := \{p = 1.0, A = 0.015545, \mathrm{alpha\_1} = 0.20548, \mathrm{beta\_1} = 14.1189, \mathrm{beta\_2} = 6.1977, \mathrm{beta\_3} = 3.3662, \mathrm{beta\_4} = 0.62517\}$$

$$\mathrm{equc7} := \mathrm{e\_c\_u\_1} = -0.031090\,(1 + 0.20548\ \mathrm{rs})\ \ln\left(1 + \frac{32.16468318}{14.1189\,\sqrt{\mathrm{rs}} + 6.1977\ \mathrm{rs} + 3.3662\ \mathrm{rs}^{(3/2)} + 0.62517\ \mathrm{rs}^{2.0}}\right)$$

```
> equc8:={p=1.0,A=0.16887,alpha_1=0.11125,beta_1=10.357,beta_2=3.6231
  ,beta_3=0.88026,
    beta_4=0.49671};
  equc9:=alpha_c=-subs(equc8,G_uc);
```

$$\text{equc8} := \{p = 1.0, A = 0.16887, \text{beta\_1} = 10.357, \text{beta\_2} = 3.6231, \text{beta\_3} = 0.88026, \text{beta\_4} = 0.49671, \text{alpha\_1} = 0.11125\}$$

$$\text{equc9} := \text{alpha\_c} = 0.33774\,(1 + 0.11125\,\text{rs})\ln\left(1 + \frac{2.960857464}{10.357\sqrt{\text{rs}} + 3.6231\,\text{rs} + 0.88026\,\text{rs}^{(3/2)} + 0.49671\,\text{rs}^{2.0}}\right)$$

```
> eqs_e_c_unif:=[eqc3,eqc7,equc5,equc7,equc9,equc3,equc2,equc1];
```

$$\text{eqs\_e\_c\_unif} := \left[\chi = \frac{\text{rhoa} - \text{rhob}}{\rho}, \text{rs} = \frac{1}{4}3^{(1/3)}4^{(2/3)}\left(\frac{1}{\pi\rho}\right)^{(1/3)}, \text{e\_c\_u\_0} = -0.062182\,(1 + 0.21370\,\text{rs})\ln\left(1 + \frac{16.08182432}{7.5957\sqrt{\text{rs}} + 3.5876\,\text{rs} + 1.6382\,\text{rs}^{(3/2)} + 0.49294\,\text{rs}^{2.0}}\right), \text{e\_c\_u\_1} = -0.031090\,(1 + 0.20548\,\text{rs})\ln\left(1 + \frac{32.16468318}{14.1189\sqrt{\text{rs}} + 6.1977\,\text{rs} + 3.3662\,\text{rs}^{(3/2)} + 0.62517\,\text{rs}^{2.0}}\right), \text{alpha\_c} = 0.33774\,(1 + 0.11125\,\text{rs})\ln\left(1 + \frac{2.960857464}{10.357\sqrt{\text{rs}} + 3.6231\,\text{rs} + 0.88026\,\text{rs}^{(3/2)} + 0.49671\,\text{rs}^{2.0}}\right), \text{f\_ii\_0} = \frac{8}{9\left(2\,2^{(1/3)} - 2\right)}, f = \frac{(1+\chi)^{(4/3)} + (1-\chi)^{(4/3)} - 2}{2\,2^{(1/3)} - 2}, \text{epsilon\_c\_unif} = \text{e\_c\_u\_0} + \frac{\text{alpha\_c}\,f\,(1 - \chi^4)}{\text{f\_ii\_0}} + (\text{e\_c\_u\_1} - \text{e\_c\_u\_0})\,f\,\chi^4\right]$$

```
> unk(eqs_e_c_unif);
```

$$\{\rho, \pi, \text{rhoa}, \text{rhob}\}$$

```
> loc(eqs_e_c_unif)intersect loc(eqs_pbec1);
```

$$\{\chi\}$$

```
> eqs_pbec1_ind:=subsop(3=NULL,eqs_pbec1):
  loc(eqs_e_c_unif)intersect loc(eqs_pbec1_ind);
```

$$\{\}$$

```
> eqs_pbec2:=[eqs_e_c_unif[i]$i=1..nops(eqs_e_c_unif),eqs_pbec1_ind[i
  ]$i=1..nops(eqs_pbec1_ind)];
```

$$\mathrm{eqs\_pbec2} := \Bigg[\chi = \frac{\mathrm{rhoa} - \mathrm{rhob}}{\rho},\ \mathrm{rs} = \frac{1}{4} 3^{(1/3)} 4^{(2/3)} \left(\frac{1}{\pi \rho}\right)^{(1/3)},\ \mathrm{e\_c\_u\_0} = -0.062182\,(1 + 0.21370\,\mathrm{rs})$$

$$\ln\left(1 + \frac{16.08182432}{7.5957 \sqrt{\mathrm{rs}} + 3.5876\,\mathrm{rs} + 1.6382\,\mathrm{rs}^{(3/2)} + 0.49294\,\mathrm{rs}^{2.0}}\right),\ \mathrm{e\_c\_u\_1} = -0.031090\,(1$$

$$+ 0.20548\,\mathrm{rs}) \ln\left(1 + \frac{32.16468318}{14.1189 \sqrt{\mathrm{rs}} + 6.1977\,\mathrm{rs} + 3.3662\,\mathrm{rs}^{(3/2)} + 0.62517\,\mathrm{rs}^{2.0}}\right),\ \mathrm{alpha\_c} =$$

$$0.33774\,(1 + 0.11125\,\mathrm{rs}) \ln\left(1 + \frac{2.960857464}{10.357 \sqrt{\mathrm{rs}} + 3.6231\,\mathrm{rs} + 0.88026\,\mathrm{rs}^{(3/2)} + 0.49671\,\mathrm{rs}^{2.0}}\right),$$

$$\mathrm{f\_ii\_0} = \frac{8}{9\left(2\,2^{(1/3)} - 2\right)},\ f = \frac{(1+\chi)^{(4/3)} + (1-\chi)^{(4/3)} - 2}{2\,2^{(1/3)} - 2},$$

$$\mathrm{epsilon\_c\_unif} = \mathrm{e\_c\_u\_0} + \frac{\mathrm{alpha\_c}\, f\,(1 - \chi^4)}{\mathrm{f\_ii\_0}} + (\mathrm{e\_c\_u\_1} - \mathrm{e\_c\_u\_0})\, f\, \chi^4,\ \beta = 0.066725,$$

$$\mathrm{gamma\_var} = \frac{1 - \ln(2)}{\pi^2},\ \phi = \frac{1}{2}(1+\chi)^{(2/3)} + \frac{1}{2}(1-\chi)^{(2/3)},\ \mathrm{k\_f} = 3^{(1/3)} (\pi^2 \rho)^{(1/3)},\ \mathrm{k\_s} = 2\sqrt{\frac{\mathrm{k\_f}}{\pi}},$$

$$t = \frac{\mathrm{norm\_drho}}{2\,\phi\,\mathrm{k\_s}\,\rho},\ A = \frac{\beta}{\mathrm{gamma\_var}\left(e^{\left(-\frac{\mathrm{epsilon\_c\_unif}}{\mathrm{gamma\_var}\,\phi^3}\right)} - 1\right)},$$

$$H = \mathrm{gamma\_var}\,\phi^3 \ln\left(1 + \frac{\beta t^2 (1 + A t^2)}{\mathrm{gamma\_var}\,(1 + A t^2 + A^2 t^4)}\right),$$

$$\mathrm{epsilon\_cGGA} = \mathrm{epsilon\_c\_unif} + H \Bigg]$$

```
> unk(eqs_pbec2);
```

$$\{\mathrm{norm\_drho}, \rho, \pi, \mathrm{rhoa}, \mathrm{rhob}\}$$

```
> loc(eqs_pbec2);
```

$$\{f, t, H, \beta, \mathrm{gamma\_var}, \mathrm{e\_c\_u\_0}, \mathrm{alpha\_c}, \mathrm{epsilon\_c\_unif}, \mathrm{e\_c\_u\_1}, \mathrm{rs}, \chi, \mathrm{k\_s}, \phi, \mathrm{f\_ii\_0}, A, \mathrm{k\_f}, \mathrm{epsilon\_cGGA}\}$$

```
> eqs pbec3:=subs(map(x->x=x|| s1,loc(eqs pbec2)),epsilon cGGA s1=eps
ilon_cGGA_1_0,rhob=0,norm_drho=norm_drhoa,
  norm_drhob=0,rho=rhoa,eqs_pbec2):
```

```
> unk(eqs_pbec3);
```

$$\{\pi, \text{rhoa}, \text{norm\_drhoa}\}$$

```
> eqs_pbec4:=subs(map(x->x=x||_s2,loc(eqs_pbec2)),epsilon_cGGA_s2=eps
ilon_cGGA_0_1,rhoa=0,norm_drho=norm_drhob,
  norm_drhoa=0,rho=rhob,eqs_pbec2):
```

```
> unk(eqs_pbec4);
```

$$\{\pi, \text{rhob}, \text{norm\_drhob}\}$$

```
> loc(eqs_pbec2)intersect loc(eqs_c1);
```

$$\{\chi\}$$

```
> eqs_c1_ind:=subsop(2=NULL,eqs_c1):
loc(eqs_pbec2)intersect loc(eqs_c1_ind);
```

$$\{\}$$

```
> eqs_c2:=[
eqs_pbec3[i]$i=1..nops(eqs_pbec3),
eqs_pbec4[i]$i=1..nops(eqs_pbec4),
eqs_pbec2[i]$i=1..nops(eqs_pbec2),
eqs_c1_ind[i]$i=1..nops(eqs_c1_ind)
]:
```

```
> unk(eqs_c2);
```

$$\{\text{norm\_drho}, \rho, \tau, \pi, \text{rhoa}, \text{rhob}, \text{norm\_drhoa}, \text{norm\_drhob}\}$$

```
>
```

LDA

```
> loc(eqs_ex_lda)intersect loc(eqs_c2);
```

$$\{\text{tau\_w}\}$$

```
> eqs_ex_lda_int:=subsop(7=NULL,eqs_ex_lda):
loc(eqs_ex_lda_int)intersect loc(eqs_c2);
```

$$\{\}$$

```
> eqs_lda:=subs(rhoa=rho/2,norm_drhoa=norm_drho/2,rhob=rho/2,norm_drhob=
norm_drho/2,
[eqs_c2[i]$i=1..nops(eqs_c2),eqs_ex_lda_int[i]$i=1..nops(eqs_ex_lda_in
t),energy=ec+ex_lda]):
```

```
> unk(eqs_lda);
```

$$\{\text{norm\_drho}, \rho, \tau, \pi\}$$

```
> def_lda:=definizioni(eqs_lda);
```

def_lda := [chi_s1, rs_s1, e_c_u_0_s1, e_c_u_1_s1, alpha_c_s1, f_ii_0_s1, f_s1, epsilon_c_unif_s1,
beta_s1, gamma_var_s1, phi_s1, k_f_s1, k_s_s1, t_s1, A_s1, H_s1, epsilon_cGGA_1_0, chi_s2,
rs_s2, e_c_u_0_s2, e_c_u_1_s2, alpha_c_s2, f_ii_0_s2, f_s2, epsilon_c_unif_s2, beta_s2,
gamma_var_s2, phi_s2, k_f_s2, k_s_s2, t_s2, A_s2, H_s2, epsilon_cGGA_0_1, χ, rs, e_c_u_0,

```
e_c_u_1, alpha_c, f_ii_0, f, epsilon_c_unif, β, gamma_var, ϕ, k_f, k_s, t, A, H, epsilon_cGGA, d,
norm_dchi, eps, C_chi, tau_w, C_chi_eps, ma, mb, epsilon_cRevPKZB, ec, κ, μ, b, c, e_var, p, z, α,
tildeq_b, x, Fx, ex_unif, ex_lda, energy]
```

```
> ima:=indice(ma,def_lda);
  imb:=indice(mb,def_lda);
```

```
ima := 58
imb := 59
```

```
> eqMa:=eqs_lda[ima];
  eqMb:=eqs_lda[imb];
```

```
eqMa := ma = max(epsilon_cGGA_1_0, epsilon_cGGA)
eqMb := mb = max(epsilon_cGGA_0_1, epsilon_cGGA)
```

```
> eqMas:=[ma=epsilon_cGGA,ma=epsilon_cGGA_1_0]:
  eqMbs:=[mb=epsilon_cGGA,mb=epsilon_cGGA_0_1]:
```

```
> calcDerivs:=proc(eqs) local cs,r,d,eq,eq2,eq3;

  cs:=CompSeq(locals=loc(eqs),globals={Pi},params=[rho,norm_drho,tau],eqs);
   r:=convert(cs,procedure);
   d:=[D[1](r),D[2](r),D[3](r)];
   eq:=map(f->op(4,convert(f,CompSeq)),d);
   # ensure that the variables are bound in the global namespace

  eq2:=map(f->evalindets(f,symbol,g->convert(convert(g,string),symbol)),eq);

  eq3:=[subs(result=deriv_rho,eq2[1]),subs(result=deriv_norm_drho,eq2[2]),
         subs(result=deriv_tau,eq2[3])];
  end proc:
```

```
> for i from 1 to 2 do
   for j from 1 to 2 do

  deriv_lda[i,j]:=calcDerivs(subsop(ima=eqMas[i],imb=eqMbs[j],eqs_lda)):
   end do;
  end do;
  i:='i':j:='j':
```

```
> ims:= eqs->select(x->x>0,[indice(ma,definizioni(eqs)),
     indice(mb,definizioni(eqs)),
     indice(marho,definizioni(eqs)),
     indice(mbrho,definizioni(eqs)),
     indice(manorm_drho,definizioni(eqs)),
     indice(mbnorm_drho,definizioni(eqs)),
     indice(matau,definizioni(eqs)),
     indice(mbtau,definizioni(eqs))]):
```

```
> eqss_lda2:=[sostConst(eqs_lda),seq(seq(seq(deriv_lda[i,j][ider],i=1..2),j=1..2),ider=1..3)]:
```

```
> checkCompatible(eqss_lda1,ims);
```

```
true
```

Order sequence defs

```
> def_eqss_lda2:=map(definizioni,eqss_lda2):
```

```
> allDefs_eqs_lda2:=combineDefs(def_eqss_lda2):
```

```
> eqs_lda2:=combineEqs(allDefs_eqs_lda2,eqss_lda2,def_eqss_lda2):
```

```
> unk(eqs_lda2);
```

```
{π, norm_drho, ρ, τ}
```

```
> getDef(ma,eqs_lda2);
  getDef(mb,eqs_lda2);
  indice(marho,eqs_lda2);
  getDef(marho,deriv_lda[1,1][1]);
  getDef(mbrho,deriv_lda[2,2][1]);
```

```
ma = max(epsilon_cGGA_1_0, epsilon_cGGA)
mb = max(epsilon_cGGA_0_1, epsilon_cGGA)
-1
marho = epsilon_cGGArho
mbrho = epsilon_cGGA_0_1rho
```

```
> a -> if a>0 then 1 else -a end if;
```

```
a → if 0 < a then 1 else -a end if;
```

```
> corrMabEqs:=proc() local arg,res,der;
  arg:=[rho,norm_drho];
  res:=[];
  for der in arg do
    res:=[op(res),
      ma||der=myF(evalb(epsilon_cGGA_1_0 >
  epsilon_cGGA),epsilon_cGGA_1_0||der,epsilon_cGGA||der),
      mb||der=myF(evalb(epsilon_cGGA_0_1 >
  epsilon_cGGA),epsilon_cGGA_0_1||der,epsilon_cGGA||der)
  ];
  end do;
  subs(myF=`if`,res);
  end proc();
```

```
corrMabEqs := [
    marho = if(epsilon_cGGA - epsilon_cGGA_1_0 < 0, epsilon_cGGA_1_0rho, epsilon_cGGArho),
    mbrho = if(epsilon_cGGA - epsilon_cGGA_0_1 < 0, epsilon_cGGA_0_1rho, epsilon_cGGArho),
    manorm_drho = if(epsilon_cGGA - epsilon_cGGA_1_0 < 0, epsilon_cGGA_1_0norm_drho,
    epsilon_cGGAnorm_drho), mbnorm_drho = if(epsilon_cGGA - epsilon_cGGA_0_1 < 0,
    epsilon_cGGA_0_1norm_drho, epsilon_cGGAnorm_drho)]
```

```
> sostCorrMabEqs:=[seq(indiceDef(lhs(corrMabEqs[i]),eqs_lda2)=(corrMabEq
  s[i]),i=1..nops(corrMabEqs))];
```

```
sostCorrMabEqs := [82 = marho = if(epsilon_cGGA - epsilon_cGGA_1_0 < 0, epsilon_cGGA_1_0rho,
    epsilon_cGGArho), 83 = mbrho = if(epsilon_cGGA - epsilon_cGGA_0_1 < 0,
    epsilon_cGGA_0_1rho, epsilon_cGGArho), 117 = manorm_drho = if(
    epsilon_cGGA - epsilon_cGGA_1_0 < 0, epsilon_cGGA_1_0norm_drho, epsilon_cGGAnorm_drho)
    118 = mbnorm_drho = if(epsilon_cGGA - epsilon_cGGA_0_1 < 0, epsilon_cGGA_0_1norm_drho,
        epsilon_cGGAnorm_drho)]
```

```
> eqs_lda3:=subsop(op(sostCorrMabEqs),eqs_lda2):
```

```
> getDef(mbrho,eqs_lda3);
      mbrho = if(epsilon_cGGA - epsilon_cGGA_0_1 < 0, epsilon_cGGA_0_1rho, epsilon_cGGArho)
```

```
> unk([op(eqs_lda3),result=deriv_rho]);
                 {π, norm_drho, ρ, τ, epsilon_cGGA_0_1rho, epsilon_cGGA_1_0rho}
```

```
> eqs_lda4:=enforceDependencies(eqs_lda3):
```

```
> unk([op(eqs_lda4),result=energy]);
  unk([op(eqs_lda4),result=deriv_rho]);
  unk([op(eqs_lda4),result=deriv_norm_drho]);
  unk([op(eqs_lda4),result=deriv_tau]);
                              {π, norm_drho, ρ, τ}

                              {π, norm_drho, ρ, τ}

                              {π, norm_drho, ρ, τ}

                              {π, norm_drho, ρ, τ}
```

```
> Fortran(eqs_lda4,defaulttype=float,optimize);
Warning, the function names {`if`} are not recognized in the target
language


Warning, The following variable name replacements were made: ["cg",
"cg0", "cg1", "cg10", "cg100", "cg101", "cg102", "cg103", "cg104",
"cg105", "cg106", "cg107", "cg11", "cg12", "cg13", "cg14", "cg15",
"cg16", "cg17", "cg18", "cg19", "cg2", "cg20", "cg21", "cg22", "cg23",
"cg24", "cg25", "cg26", "cg27", "cg28", "cg29", "cg3", "cg30", "cg31",
"cg32", "cg33", "cg34", "cg35", "cg36", "cg37", "cg38", "cg39", "cg4",
"cg40", "cg41", "cg42", "cg43", "cg44", "cg45", "cg46", "cg47", "cg48",
"cg49", "cg5", "cg50", "cg51", "cg52", "cg53", "cg54", "cg55", "cg56",
"cg57", "cg58", "cg59", "cg6", "cg60", "cg61", "cg62", "cg63", "cg64",
"cg65", "cg66", "cg67", "cg68", "cg69", "cg7", "cg70", "cg71", "cg72",
"cg73", "cg74", "cg75", "cg76", "cg77", "cg78", "cg79", "cg8", "cg80",
"cg81", "cg82", "cg83", "cg84", "cg85", "cg86", "cg87", "cg88", "cg89",
"cg9", "cg90", "cg91", "cg92", "cg93", "cg94", "cg95", "cg96", "cg97",
"cg98", "cg99"] = ["chi_s1", "rs_s1", "e_c_u_0_s1", "k_f_s1",
"epsilon_cRevPKZBnorm_drho", "ecnorm_drho", "deriv_norm_drho",
"epsilon_cRevPKZBtau", "alphatau", "tildeq_btau", "ex_ldatau",
"deriv_tau", "k_s_s1", "norm_drho", "t_s1", "A_s1", "H_s1",
"epsilon_cGGA_1_0", "chi_s2", "rs_s2", "e_c_u_0_s2", "e_c_u_1_s1",
"e_c_u_1_s2", "alpha_c_s2", "f_ii_0_s2", "f_s2", "epsilon_c_unif_s2",
"beta_s2", "gamma_var_s2", "phi_s2", "k_f_s2", "k_s_s2", "alpha_c_s1",
"t_s2", "A_s2", "H_s2", "epsilon_cGGA_0_1", "e_c_u_0", "e_c_u_1",
"alpha_c", "f_ii_0", "epsilon_c_unif", "gamma_var", "f_ii_0_s1", "k_f",
"k_s", "epsilon_cGGA", "C_chi", "tau_w", "C_chi_eps",
"epsilon_cRevPKZB", "e_var", "tildeq_b", "ex_unif", "f_s1", "ex_lda",
"e_c_u_0rho", "epsilon_c_unifrho", "k_frho", "k_srho",
"epsilon_cGGArho", "tau_wrho", "alpharho", "tildeq_brho", "ex_unifrho",
"epsilon_c_unif_s1", "ex_ldarho", "rs_s1rho", "e_c_u_1_s1rho",
"epsilon_c_unif_s1rho", "k_f_s1rho", "k_s_s1rho", "t_s1rho", "A_s1rho",
"H_s1rho", "epsilon_cGGA_1_0rho", "beta_s1", "rs_s2rho",
"e_c_u_1_s2rho", "epsilon_c_unif_s2rho", "k_f_s2rho", "k_s_s2rho",
"t_s2rho", "A_s2rho", "H_s2rho", "epsilon_cGGA_0_1rho",
"epsilon_cRevPKZBrho", "gamma_var_s1", "deriv_rho", "tnorm_drho",
"Hnorm_drho", "epsilon_cGGAnorm_drho", "tau_wnorm_drho", "pnorm_drho",
"znorm_drho", "alphanorm_drho", "tildeq_bnorm_drho", "xnorm_drho",
"phi_s1", "Fxnorm_drho", "ex_ldanorm_drho", "t_s1norm_drho",
"H_s1norm_drho", "epsilon_cGGA_1_0norm_drho", "manorm_drho",
"t_s2norm_drho", "H_s2norm_drho", "epsilon_cGGA_0_1norm_drho",
"mbnorm_drho"]
```

```
      cg = 1
      t1 = 3 ** (0.1D1 / 0.3D1)
      t2 = 4 ** (0.1D1 / 0.3D1)
      t3 = t2 ** 2
      t4 = t1 * t3
      t5 = 2 ** (0.1D1 / 0.3D1)
      t6 = 0.1D1 / 0.3141592654D1
      t7 = 0.1D1 / rho
      t9 = (t6 * t7) ** (0.1D1 / 0.3D1)
      cg0 = dble(t4) * dble(t5) * t9 / 0.4D1
      t14 = sqrt(cg0)
      t17 = t14 * cg0
      t19 = cg0 ** 0.20D1
      t25 = log(0.1D1 + 0.1608182432D2 / (0.75957D1 * t14 + 0.35876D1 *
     #cg0 + 0.16382D1 * t17 + 0.49294D0 * t19))
      cg1 = -0.62182D-1 * (0.1D1 + 0.21370D0 * cg0) * t25
      t29 = 0.1D1 + 0.20548D0 * cg0
      t34 = 0.141189D2 * t14 + 0.61977D1 * cg0 + 0.33662D1 * t17 + 0.625
     #17D0 * t19
      t37 = 0.1D1 + 0.3216468318D2 / t34
      t38 = log(t37)
      cg2 = -0.31090D-1 * t29 * t38
      t51 = log(0.1D1 + 0.2960857464D1 / (0.10357D2 * t14 + 0.36231D1 *
     #cg0 + 0.88026D0 * t17 + 0.49671D0 * t19))
      cg3 = 0.33774D0 * (0.1D1 + 0.11125D0 * cg0) * t51
      cg4 = 0.8D1 / 0.9D1 / dble(2 * t5 - 2)
      cg5 = 1
      cg6 = cg2
      cg7 = 0.66725D-1
      t56 = log(0.2D1)
      t58 = 0.3141592654D1 ** 2
      cg8 = (0.1D1 - t56) / t58
      t60 = t5 ** 2
      cg9 = dble(t60) / 0.2D1
      t61 = t1 * t60
      t63 = (t58 * rho) ** (0.1D1 / 0.3D1)
      cg10 = dble(t61) * t63 / 0.2D1
      t66 = sqrt(cg10 * t6)
      cg11 = 0.2D1 * t66
      t67 = 0.1D1 / cg9
      t68 = cg12 * t67
      t69 = 0.1D1 / cg11
      cg13 = t68 * t69 * t7 / 0.2D1
      t72 = 0.1D1 / cg8
      t74 = cg9 ** 2
      t75 = t74 * cg9
      t76 = 0.1D1 / t75
      t78 = exp(-cg2 * t72 * t76)
      t79 = t78 - 0.1D1
      cg14 = 0.66725D-1 * t72 / t79
      t82 = cg8 * t75
      t83 = cg13 ** 2
      t84 = t72 * t83
      t85 = cg14 * t83
      t86 = 0.1D1 + t85
      t87 = cg14 ** 2
      t88 = t83 ** 2
      t90 = 0.1D1 + t85 + t87 * t88
      t91 = 0.1D1 / t90
      t92 = t86 * t91
      t95 = 0.1D1 + 0.66725D-1 * t84 * t92
      t96 = log(t95)
      cg15 = t82 * t96
      cg16 = cg2 + cg15
      cg17 = -1
      cg18 = cg0
      t99 = sqrt(cg18)
      t102 = t99 * cg18
      t104 = cg18 ** 0.20D1
      t110 = log(0.1D1 + 0.1608182432D2 / (0.75957D1 * t99 + 0.35876D1 *
     # cg18 + 0.16382D1 * t102 + 0.49294D0 * t104))
      cg19 = -0.62182D-1 * (0.1D1 + 0.21370D0 * cg18) * t110
      t114 = 0.1D1 + 0.20548D0 * cg18
      t119 = 0.141189D2 * t99 + 0.61977D1 * cg18 + 0.33662D1 * t102 + 0.
     #62517D0 * t104
      t122 = 0.1D1 + 0.3216468318D2 / t119
      t123 = log(t122)
      cg20 = -0.31090D-1 * t114 * t123
      t136 = log(0.1D1 + 0.2960857464D1 / (0.10357D2 * t99 + 0.36231D1 *
     # cg18 + 0.88026D0 * t102 + 0.49671D0 * t104))
      cg21 = 0.33774D0 * (0.1D1 + 0.11125D0 * cg18) * t136
      cg22 = cg4
      cg23 = 1
      cg24 = cg20
      cg25 = 0.66725D-1
      cg26 = cg8
      cg27 = cg9
      cg28 = cg10
      t139 = sqrt(cg28 * t6)
      cg29 = 0.2D1 * t139
      t140 = 0.1D1 / cg27
      t141 = cg12 * t140
      t142 = 0.1D1 / cg29
      cg30 = t141 * t142 * t7 / 0.2D1
      t145 = 0.1D1 / cg26
      t147 = cg27 ** 2
      t148 = t147 * cg27
      t149 = 0.1D1 / t148
      t151 = exp(-cg20 * t145 * t149)
      t152 = t151 - 0.1D1
      cg31 = 0.66725D-1 * t145 / t152
      t155 = cg26 * t148
      t156 = cg30 ** 2
      t157 = t145 * t156
      t158 = cg31 * t156
      t159 = 0.1D1 + t158
      t160 = cg31 ** 2
      t161 = t156 ** 2
      t163 = 0.1D1 + t158 + t160 * t161
      t164 = 0.1D1 / t163
      t165 = t159 * t164
      t168 = 0.1D1 + 0.66725D-1 * t157 * t165
      t169 = log(t168)
      cg32 = t155 * t169
      cg33 = cg20 + cg32
      rs = dble(t4) * t9 / 0.4D1
      t172 = 0.1D1 + 0.21370D0 * rs
      t173 = sqrt(rs)
      t176 = t173 * rs
      t178 = rs ** 0.20D1
      t180 = 0.75957D1 * t173 + 0.35876D1 * rs + 0.16382D1 * t176 + 0.49
     #294D0 * t178
      t183 = 0.1D1 + 0.1608182432D2 / t180
      t184 = log(t183)
      cg34 = -0.62182D-1 * t172 * t184
      t197 = log(0.1D1 + 0.3216468318D2 / (0.141189D2 * t173 + 0.61977D1
     # * rs + 0.33662D1 * t176 + 0.62517D0 * t178))
      cg35 = -0.31090D-1 * (0.1D1 + 0.20548D0 * rs) * t197
      t210 = log(0.1D1 + 0.2960857464D1 / (0.10357D2 * t173 + 0.36231D1
     #* rs + 0.88026D0 * t176 + 0.49671D0 * t178))
      cg36 = 0.33774D0 * (0.1D1 + 0.11125D0 * rs) * t210
      cg37 = cg22
      cg38 = cg34
      beta = 0.66725D-1
      cg39 = cg26
      phi = 1
      cg40 = dble(t1) * t63
      t213 = sqrt(cg40 * t6)
      cg41 = 0.2D1 * t213
      t214 = 0.1D1 / cg41
      t215 = cg12 * t214
      t = t215 * t7 / 0.2D1
      t217 = 0.1D1 / cg39
      t219 = exp(-cg34 * t217)
      t220 = -0.1D1 + t219
      A = 0.66725D-1 * t217 / t220
      t223 = t ** 2
      t224 = t217 * t223
      t225 = A * t223
      t226 = 0.1D1 + t225
      t227 = A ** 2
      t228 = t223 ** 2
      t230 = 0.1D1 + t225 + t227 * t228
      t231 = 0.1D1 / t230
      t232 = t226 * t231
      t235 = 0.1D1 + 0.66725D-1 * t224 * t232
      t236 = log(t235)
      H = cg39 * t236
      cg42 = cg34 + H
      d = 0.28D1
      cg43 = 0.53D0
      t237 = cg12 ** 2
      cg44 = t237 * t7 / 0.8D1
      cg45 = 0.53D0
      ma = max(cg16, cg42)
      mb = max(cg33, cg42)
      t239 = cg44 ** 2
      t240 = tau ** 2
      t241 = 0.1D1 / t240
      t242 = t239 * t241
      t244 = 0.1D1 + 0.53D0 * t242
      t247 = ma / 0.2D1 + mb / 0.2D1
      cg46 = cg42 * t244 - 0.153D1 * t242 * t247
      t250 = rho * cg46
      t251 = t239 * cg44
      t252 = cg46 * t251
      t254 = 0.1D1 / t240 / tau
      t257 = 0.1D1 + 0.28D1 * t252 * t254
      ec = t250 * t257
      kappa = 0.804D0
      mu = 0.21951D0
      b = 0.4D0
      c = 0.159096D1
      cg47 = 0.1537D1
      t258 = t237 * dble(t1)
      t259 = t58 ** (0.1D1 / 0.3D1)
      t260 = t259 ** 2
      t261 = 0.1D1 / t260
      t262 = rho ** 2
      t263 = rho ** (0.1D1 / 0.3D1)
      t264 = t263 ** 2
      t267 = t261 / t264 / t262
      p = t258 * t267 / 0.12D2
      t269 = 0.1D1 / tau
      z = cg44 * t269
      t271 = 0.1D1 / z - 0.1D1
      alpha = 0.5D1 / 0.3D1 * p * t271
      t273 = alpha - 0.1D1
      t276 = 0.1D1 + 0.4D0 * alpha * t273
      t277 = sqrt(t276)
      t278 = 0.1D1 / t277
      cg48 = 0.9D1 / 0.20D2 * t273 * t278 + 0.2D1 / 0.3D1 * p
      t282 = z ** 2
      t283 = 0.1D1 + t282
      t284 = t283 ** 2
      t285 = 0.1D1 / t284
      t288 = 0.10D2 / 0.81D2 + 0.159096D1 * t282 * t285
      t290 = cg48 ** 2
      t293 = p ** 2
      t296 = sqrt(0.18D2 * t282 + 0.50D2 * t293)
      t303 = t288 * p + 0.146D3 / 0.2025D4 * t290 - 0.73D2 / 0.4050D4 *
     #cg48 * t296 + 0.1895718785D-1 * t293 + 0.1102007148D0 * t282 + 0.3
     #3738687D0 * t293 * p
      t305 = 0.1D1 + 0.1239758041D1 * p
      t306 = t305 ** 2
      t307 = 0.1D1 / t306
      x = t303 * t307
      t309 = 0.1D1 + 0.1243781095D1 * x
      Fx = 0.1804D1 - 0.804D0 / t309
      cg49 = -0.3D1 / 0.4D1 * t6 * dble(t1) * t63
      t315 = rho * cg49
      cg50 = t315 * Fx
      energy = ec + cg50
      t316 = t9 ** 2
      t319 = 0.1D1 / t262
      t320 = 0.1D1 / t316 * t6 * t319
      rsrho = -dble(t4) * t320 / 0.12D2
      t325 = t180 ** 2
      t334 = rs ** 0.10D1
      cg51 = -0.1328829340D-1 * rsrho * t184 + 0.9999999999D0 * t172 / t
     #325 * (0.3797850000D1 / t173 * rsrho + 0.35876D1 * rsrho + 0.24573
     #00000D1 * t173 * rsrho + 0.985880D0 * t334 * rsrho) / t183
      cg52 = cg51
      t342 = t63 ** 2
      t343 = 0.1D1 / t342
      cg53 = dble(t1) * t343 * t58 / 0.3D1
      cg54 = 0.1D1 / t213 * cg53 * t6
      t348 = cg41 ** 2
      trho = -cg12 / t348 * t7 * cg54 / 0.2D1 - t215 * t319 / 0.2D1
      t355 = cg39 ** 2
      t357 = t220 ** 2
      Arho = 0.66725D-1 / t355 / t357 * cg51 * t219
      t362 = t217 * t
      t366 = Arho * t223
      t367 = A * t
      t369 = 0.2D1 * t367 * trho
      t374 = t230 ** 2
      t376 = t226 / t374
      t380 = t223 * t
      t381 = t227 * t380
      t390 = 0.1D1 / t235
      Hrho = cg39 * (0.133450D0 * t362 * t232 * trho + 0.66725D-1 * t224
     # * (t366 + t369) * t231 - 0.66725D-1 * t224 * t376 * (t366 + t369
     #+ 0.2D1 * A * t228 * Arho + 0.4D1 * t381 * trho)) * t390
      cg55 = cg51 + Hrho
      cg56 = -t237 * t319 / 0.8D1
      prho = -0.2D1 / 0.9D1 * t258 * t261 / t264 / t262 / rho
      zrho = cg56 * t269
      t401 = p / t282
      cg57 = 0.5D1 / 0.3D1 * prho * t271 - 0.5D1 / 0.3D1 * t401 * zrho
      t408 = t273 / t277 / t276
      cg58 = 0.9D1 / 0.20D2 * cg57 * t278 - 0.9D1 / 0.40D2 * t408 * (0.4
     #D0 * cg57 * t273 + 0.4D0 * alpha * cg57) + 0.2D1 / 0.3D1 * prho
      t417 = z * t285
      t423 = t282 * z / t284 / t283
      t434 = cg48 / t296
      t435 = z * zrho
      t437 = p * prho
      t450 = t303 / t306 / t305
      xrho = ((0.318192D1 * t417 * zrho - 0.636384D1 * t423 * zrho) * p
     #+ t288 * prho + 0.292D3 / 0.2025D4 * cg48 * cg58 - 0.73D2 / 0.4050
     #D4 * cg58 * t296 - 0.73D2 / 0.8100D4 * t434 * (0.36D2 * t435 + 0.1
     #00D3 * t437) + 0.3791437570D-1 * t437 + 0.2204014296D0 * t435 + 0.
     #101216061D1 * t293 * prho) * t307 - 0.2479516082D1 * t450 * prho
      t453 = t309 ** 2
      t454 = 0.1D1 / t453
      Fxrho = 0.1000000000D1 * t454 * xrho
      cg59 = -0.3141592654D1 * dble(t1) * t343 / 0.4D1
      cg60 = cg49 * Fx + rho * cg59 * Fx + t315 * Fxrho
      cg61 = -dble(t4) * dble(t5) * t320 / 0.12D2
      t468 = t34 ** 2
      t477 = cg0 ** 0.10D1
      cg62 = -0.638837320D-2 * cg61 * t38 + 0.1000000000D1 * t29 / t468
     #* (0.7059450000D1 / t14 * cg61 + 0.61977D1 * cg61 + 0.5049300000D1
     # * t14 * cg61 + 0.1250340D1 * t477 * cg61) / t37
      cg63 = cg62
      cg64 = dble(t61) * t343 * t58 / 0.6D1
      cg65 = 0.1D1 / t66 * cg64 * t6
      t489 = cg11 ** 2
      cg66 = -t68 / t489 * t7 * cg65 / 0.2D1 - t68 * t69 * t319 / 0.2D1
      t497 = cg8 ** 2
      t499 = t79 ** 2
      cg67 = 0.66725D-1 / t497 / t499 * cg62 * t76 * t78
      t505 = t72 * cg13
      t509 = cg67 * t83
      t510 = cg14 * cg13
      t512 = 0.2D1 * t510 * cg66
      t517 = t90 ** 2
      t519 = t86 / t517
      t523 = t83 * cg13
      t524 = t87 * t523
      t532 = 0.1D1 / t95
      cg68 = t82 * (0.133450D0 * t505 * t92 * cg66 + 0.66725D-1 * t84 *
     #(t509 + t512) * t91 - 0.66725D-1 * t84 * t519 * (t509 + t512 + 0.2
     #D1 * cg14 * t88 * cg67 + 0.4D1 * t524 * cg66)) * t532
      cg69 = cg62 + cg68
      t535 = cg42 - cg16 .lt. 0.0D0
      marho = if(t535, cg69, cg55)
      cg70 = cg61
      t538 = t119 ** 2
      t547 = cg18 ** 0.10D1
      cg71 = -0.638837320D-2 * cg70 * t123 + 0.1000000000D1 * t114 / t53
     #8 * (0.7059450000D1 / t99 * cg70 + 0.61977D1 * cg70 + 0.5049300000
     #D1 * t99 * cg70 + 0.1250340D1 * t547 * cg70) / t122
      cg72 = cg71
      cg73 = cg64
      cg74 = 0.1D1 / t139 * cg73 * t6
      t557 = cg29 ** 2
      cg75 = -t141 / t557 * t7 * cg74 / 0.2D1 - t141 * t142 * t319 / 0.2
     #D1
      t565 = cg26 ** 2
      t567 = t152 ** 2
      cg76 = 0.66725D-1 / t565 / t567 * cg71 * t149 * t151
      t573 = t145 * cg30
      t577 = cg76 * t156
      t578 = cg31 * cg30
      t580 = 0.2D1 * t578 * cg75
      t585 = t163 ** 2
      t587 = t159 / t585
      t591 = t156 * cg30
      t592 = t160 * t591
      t600 = 0.1D1 / t168
      cg77 = t155 * (0.133450D0 * t573 * t165 * cg75 + 0.66725D-1 * t157
     # * (t577 + t580) * t164 - 0.66725D-1 * t157 * t587 * (t577 + t580
     #+ 0.2D1 * cg31 * t161 * cg76 + 0.4D1 * t592 * cg75)) * t600
      cg78 = cg71 + cg77
      t603 = cg42 - cg33 .lt. 0.0D0
      mbrho = if(t603, cg78, cg55)
      t605 = cg42 * cg44
      t609 = cg44 * t241
      cg79 = cg55 * t244 + 0.106D1 * t605 * t241 * cg56 - 0.306D1 * t609
     # * t247 * cg56 - 0.153D1 * t242 * (marho / 0.2D1 + mbrho / 0.2D1)
      t623 = cg46 * t239
      ecrho = cg46 * t257 + rho * cg79 * t257 + t250 * (0.28D1 * cg79 *
     #t251 * t254 + 0.84D1 * t623 * t254 * cg56)
      cg80 = ecrho + cg60
      cg81 = t214 * t7 / 0.2D1

      cg82 = cg39 * (0.133450D0 * t362 * t232 * cg81 + 0.133450D0 * t217
     # * t380 * A * cg81 * t231 - 0.66725D-1 * t224 * t376 * (0.2D1 * t3
     #67 * cg81 + 0.4D1 * t381 * cg81)) * t390
      cg83 = cg82
      cg84 = cg12 * t7 / 0.4D1
      cg85 = cg12 * dble(t1) * t267 / 0.6D1
      cg86 = cg84 * t269
      cg87 = 0.5D1 / 0.3D1 * cg85 * t271 - 0.5D1 / 0.3D1 * t401 * cg86
      cg88 = 0.9D1 / 0.20D2 * cg87 * t278 - 0.9D1 / 0.40D2 * t408 * (0.4
     #D0 * cg87 * t273 + 0.4D0 * alpha * cg87) + 0.2D1 / 0.3D1 * cg85
      t675 = z * cg86
      t677 = p * cg85
      cg89 = ((0.318192D1 * t417 * cg86 - 0.636384D1 * t423 * cg86) * p
     #+ t288 * cg85 + 0.292D3 / 0.2025D4 * cg48 * cg88 - 0.73D2 / 0.4050
     #D4 * cg88 * t296 - 0.73D2 / 0.8100D4 * t434 * (0.36D2 * t675 + 0.1
     #00D3 * t677) + 0.3791437570D-1 * t677 + 0.2204014296D0 * t675 + 0.
     #101216061D1 * t293 * cg85) * t307 - 0.2479516082D1 * t450 * cg85
      cg90 = 0.1000000000D1 * t454 * cg89
      cg91 = t315 * cg90
      cg92 = t67 * t69 * t7 / 0.2D1
      cg93 = t82 * (0.133450D0 * t505 * t92 * cg92 + 0.133450D0 * t72 *
     #t523 * cg14 * cg92 * t91 - 0.66725D-1 * t84 * t519 * (0.2D1 * t510
     # * cg92 + 0.4D1 * t524 * cg92)) * t532
      cg94 = cg93
      cg95 = if(t535, cg93, cg82)
      cg96 = t140 * t142 * t7 / 0.2D1
      cg97 = t155 * (0.133450D0 * t573 * t165 * cg96 + 0.133450D0 * t145
     # * t591 * cg31 * cg96 * t164 - 0.66725D-1 * t157 * t587 * (0.2D1 *
     # t578 * cg96 + 0.4D1 * t592 * cg96)) * t600
      cg98 = cg97
      cg99 = if(t603, cg97, cg82)
      cg100 = cg82 * t244 + 0.106D1 * t605 * t241 * cg84 - 0.306D1 * t60
     #9 * t247 * cg84 - 0.153D1 * t242 * (cg95 / 0.2D1 + cg99 / 0.2D1)
      cg101 = rho * cg100 * t257 + t250 * (0.28D1 * cg100 * t251 * t254
     #+ 0.84D1 * t623 * t254 * cg84)
      cg102 = cg101 + cg91
      cg103 = -0.106D1 * cg42 * t239 * t254 + 0.306D1 * t239 * t254 * t2
     #47
      t763 = t240 ** 2
      ectau = rho * cg103 * t257 + t250 * (0.28D1 * cg103 * t251 * t254
     #- 0.84D1 * t252 / t763)
      ztau = -t609
      cg104 = -0.5D1 / 0.3D1 * t401 * ztau
      cg105 = 0.9D1 / 0.20D2 * cg104 * t278 - 0.9D1 / 0.40D2 * t408 * (0
     #.4D0 * cg104 * t273 + 0.4D0 * alpha * cg104)
      t790 = z * ztau
      xtau = ((0.318192D1 * t417 * ztau - 0.636384D1 * t423 * ztau) * p
     #+ 0.292D3 / 0.2025D4 * cg48 * cg105 - 0.73D2 / 0.4050D4 * cg105 *
     #t296 - 0.73D2 / 0.225D3 * t434 * t790 + 0.2204014296D0 * t790) * t
     #307
      Fxtau = 0.1000000000D1 * t454 * xtau
      cg106 = t315 * Fxtau
      cg107 = ectau + cg106
```

```
>
```

Tests

```
> eqFx1:=z=solve(eqx8,z);
```

$$\mathrm{eqFx1} := z = \frac{5\,p}{3\,\alpha + 5\,p}$$

```
> eqs_Fx1:=[eqk1,eqk2,eqk3,eqk4,p=s^2,eqFx1,eqx7,eqk5,eqx10,eqx9];
```

$$\mathrm{eqs\_Fx1} := \Big[\kappa = 0.804,\ \mu = 0.21951,\ b = 0.4,\ c = 1.59096,\ p = s^{2},\ z = \frac{5\,p}{3\,\alpha + 5\,p},$$

$$\mathrm{tildeq\_b} = \frac{9\,(\alpha - 1)}{20\,\sqrt{1 + b\,\alpha\,(\alpha - 1)}} + \frac{2}{3}\,p,\ \mathrm{e\_var} = 1.537,\ x = \frac{1}{\left(1 + \sqrt{\mathrm{e\_var}}\,p\right)^2}\left(\left(\frac{10}{81} + \frac{c\,z^2}{(1 + z^2)^2}\right)p + \frac{146}{2025}\,\mathrm{tildeq\_b}^2 - \frac{73}{4050}\,\mathrm{tildeq\_b}\,\sqrt{18\,z^2 + 50\,p^2} + \frac{100\,p^2}{6561\,\kappa} + \frac{4}{45}\,\sqrt{\mathrm{e\_var}}\,z^2 + \mathrm{e\_var}\,\mu\,p^3\right),$$

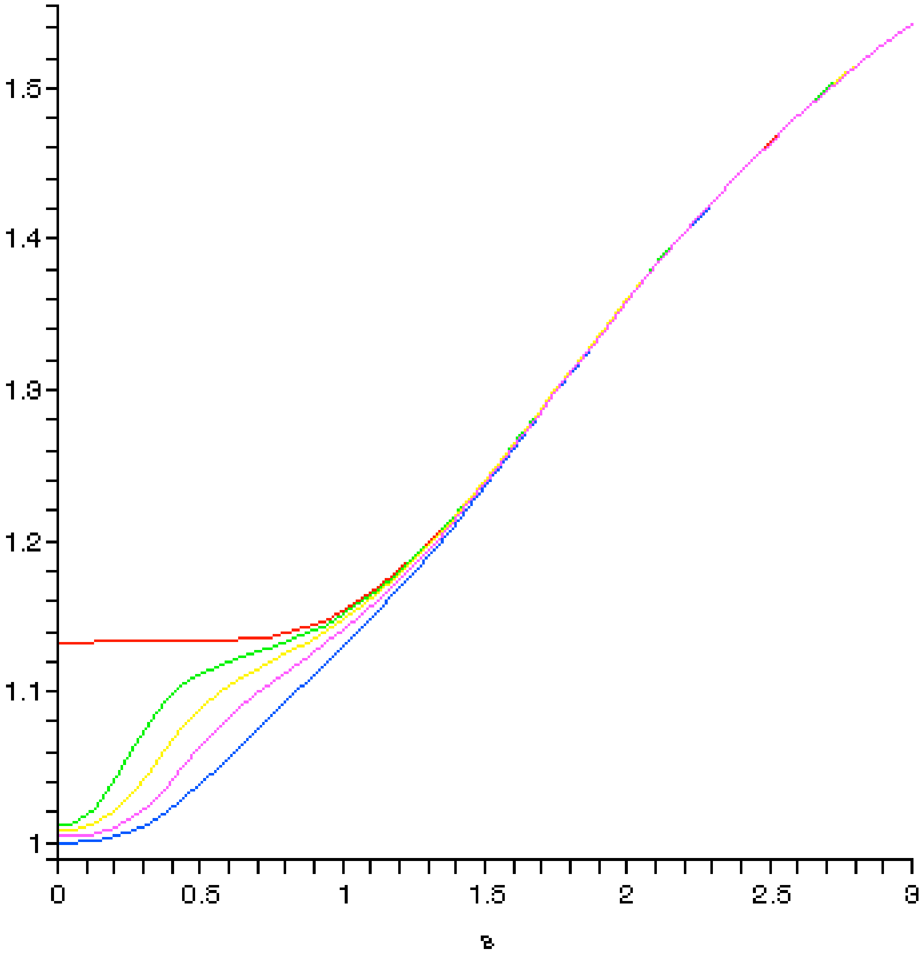
$$\left.\mathrm{Fx} = 1 + \kappa - \frac{\kappa}{1 + \frac{x}{\kappa}}\right]$$

```
> unk(eqs_Fx1);
```

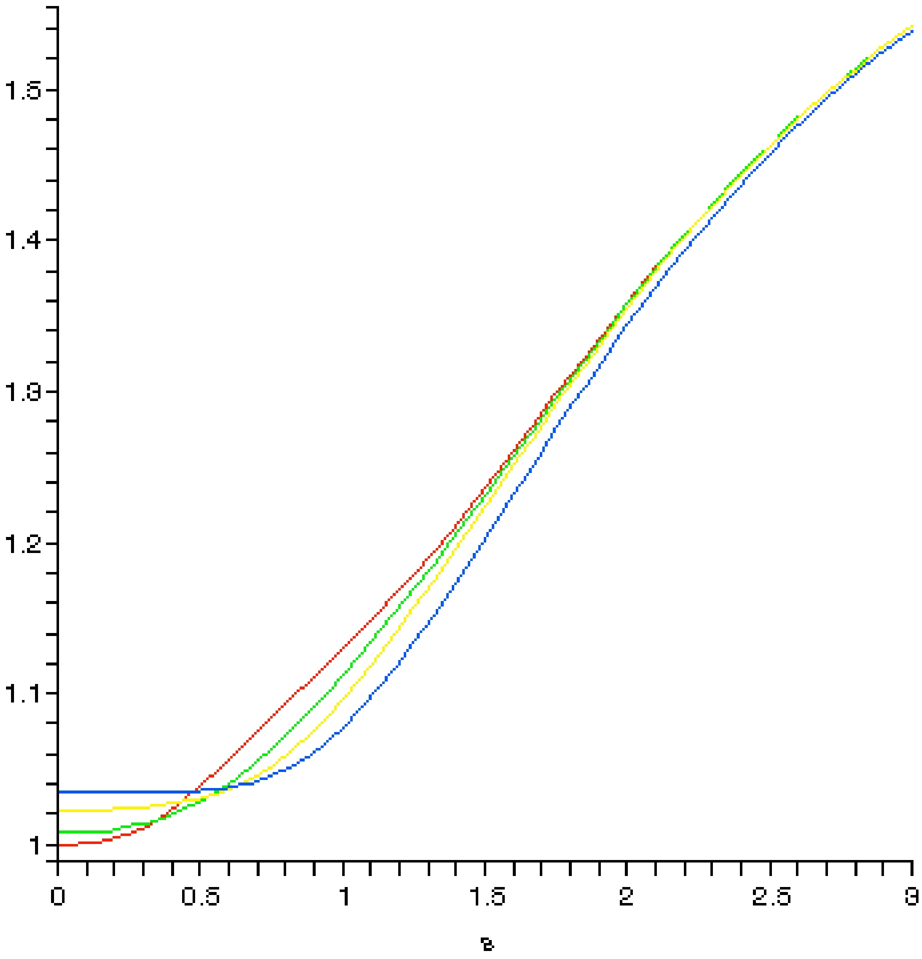
$$\{s, \alpha\}$$

```
> r_Fx1:=convert(CompSeq(locals=loc(eqs_Fx1),params=[s,alpha],eqs_Fx1),p
rocedure):
```

```
> plot({r_Fx1(s,0),r_Fx1(s,0.1),r_Fx1(s,0.25),r_Fx1(s,0.5),r_Fx1(s,1)},s
=0..3);
```



```
> plot({r_Fx1(s,1),r_Fx1(s,2),r_Fx1(s,4),r_Fx1(s,10^6)},s=0..3);
```



```
> evalf(1/infinity);
```

$$0.$$

```
> indets(collect(sost(subsop(indiceDef(ex_unif,eqs_lda4)=NULL,indiceDef(
tau_w,eqs_lda4)=NULL,indiceDef(z,eqs_lda4)=NULL,indiceDef(p,eqs_lda4)=
NULL,[op(eqs_lda4),result=energy/rho/ex_unif])),rho),symbol);
```

$$\{\pi, \mathrm{norm\_drho}, \mathrm{ex\_unif}, \rho, \mathrm{p}, \mathrm{z}, \tau, \mathrm{tau\_w}\}$$

```
> getDef(rs,eqs_lda4);
```

$$\mathrm{rs} = \frac{1}{4} 3^{(1/3)} 4^{(2/3)} \left(\frac{1}{\pi \rho}\right)^{(1/3)}$$

```
> eq_G1:=rho=solve(getDef(rs,eqs_lda4),rho);
```

$$\mathrm{eq\_G1} := \rho = \frac{3}{4 \mathrm{rs}^3 \pi}$$

```
> eqx4;
```

$$s = \frac{3^{(2/3)} \left(\frac{1}{\pi}\right)^{(2/3)} \text{norm\_drho}}{6\,\rho^{(4/3)}}$$

```
> eq_G2:=norm_drho=solve(eqx4,norm_drho);
```

$$\text{eq\_G2} := \text{norm\_drho} = \frac{2\,s\,\rho^{(4/3)}\,3^{(1/3)}}{\left(\frac{1}{\pi}\right)^{(2/3)}}$$

```
> getDef(alpha,eqs_lda4);
```

$$\alpha = \frac{5}{3}\,p\left(\frac{1}{z} - 1\right)$$

```
> getDef(z,eqs_lda4);
```

$$z = \frac{\text{tau\_w}}{\tau}$$

```
> getDef(tau_w,eqs_lda4);
```

$$\text{tau\_w} = \frac{\text{norm\_drho}^{2}}{8\,\rho}$$

```
> eq_G3_1:=subs(getDef(z,eqs_lda4),getDef(tau_w,eqs_lda4),p=s^2,getDef(alpha,eqs_lda4));
```

$$\text{eq\_G3\_1} := \alpha = \frac{5}{3}\,s^{2}\left(\frac{8\,\rho\,\tau}{\text{norm\_drho}^{2}} - 1\right)$$

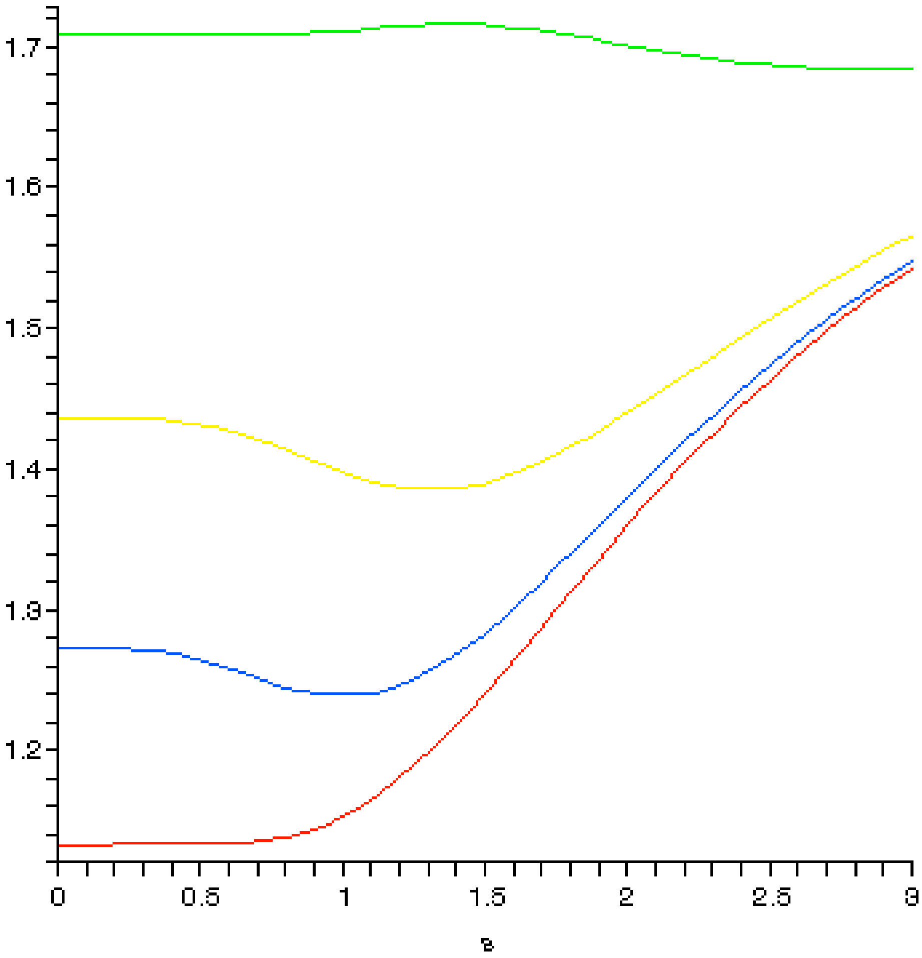
```
> eq_G3:=tau=solve(eq_G3_1,tau);
```

$$\text{eq\_G3} := \tau = \frac{(3\,\alpha + 5\,s^{2})\,\text{norm\_drho}^{2}}{40\,s^{2}\,\rho}$$

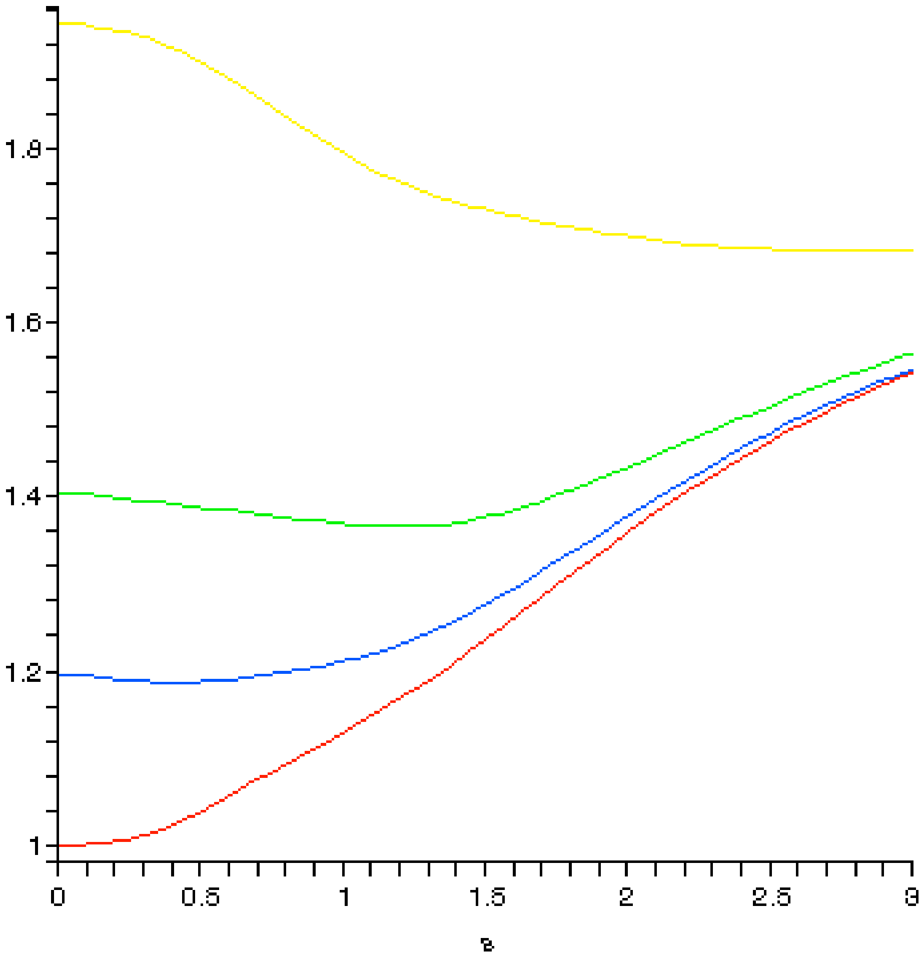
```
> eqs_G1:=[eq_G1,eq_G2,eq_G3,op(subs(s=tmp_s,alpha=tmp_alpha,rs=tmp_rs,eqs_lda4)),result=energy/(rho*ex_unif)]:
> unk(eqs_G1);
```

$$\{\pi, s, \alpha, rs\}$$

```
> cs_G1:=CompSeq(locals=loc(eqs_G1),params=[s,alpha,rs],eqs_G1):
> r_G1:=convert(cs_G1,procedure):
> plot({r_G1(s,0,0.0000001),r_G1(s,0,2),r_G1(s,0,10),r_G1(s,0,10^6)},s=0..3);
```



```
> plot({r_G1(s,1,0.0000001),r_G1(s,1,2),r_G1(s,1,10),r_G1(s,1,10^6)},s=0
  ..3);
```


1.8
1.6
1.4
1.2
1
0
0.5
1
1.5
2
2.5
3
a

>