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 (http://www.gnu.org/copyleft/gpl.html) or higher license.

"Progress in science through free software" ;)

Fawzi Mohamed

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

```
> 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':myIF:='myIF':
```

```
> 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,attComp;
    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(eqss[j]);

  #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]):
        attComp:=sameNameSameDef(eqd1,eqd2);
        res:=attComp and res;
        if not attComp then
          print("incompatibility between",i,j);
        end if;
      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:
```

```
> calcDerivs:=proc(eqs,arg_names) local cs,r,d,eq,eq2,eq3,i;
 cs:=CompSeq(locals=loc(eqs),globals=convert(unk(eqs),set)minus
convert(arg_names,set),
  params=arg_names,eqs);
 r:=convert(cs,procedure);
 d:=[seq(D[i](r),i=1..nops(arg_names))];
 eq:=map(f->op(-1,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)),e
q);

eq3:=[seq(subs(result=deriv_||(arg_names[i]),eq2[i]),i=1..nops(arg_name
s))];
end proc:
```

```
> with(CodeGeneration);
```

```
Warning, the protected name Matlab has been redefined and unprotected
```

[*C*, *Fortran*, *IntermediateCode*, *Java*, *LanguageDefinition*, *Matlab*, *Names*, *Save*, *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*tildeq
  _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];
```

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

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

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

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

$$\left.ex\_lda = \rho\ ex\_unif\,Fx\right]$$

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

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

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

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

correlation

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

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

```
> eqc2_1:=ma=max(epsilon_cGGA_1_0,epsilon_cGGA);
  eqc2_2:=mb=max(epsilon_cGGA_0_1,epsilon_cGGA);
```

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

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

```
> eqc2_3:=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);
  eqc2:=collect(eqc2_3,tau);
```

$$eqc2\_3 := epsilon\_cRevPKZB = epsilon\_cGGA\left(1 + \frac{C\_chi\_eps\ tau\_w^{2}}{\tau^{2}}\right) - \frac{(1 + C\_chi\_eps)\ tau\_w^{2}\left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho}\right)}{\tau^{2}}$$

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

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

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

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

```
> 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;
```

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

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

```
> 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;
```

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

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

$$eqs\_c1 := \left[d = 2.8, \chi = \frac{rhoa - rhob}{\rho}, norm\_dchi = \frac{1}{\rho^2}\,(2\ \mathrm{sqrt}(norm\_drhoa^2\ rhob^2\right.$$
$$+ norm\_drhob^2\ rhoa^2 - rhoa\ rhob\ norm\_drho^2 + rhoa\ rhob\ norm\_drhoa^2$$
$$+ rhoa\ rhob\ norm\_drhob^2)), eps = \frac{norm\_dchi\ 3^{(2/3)}}{6\ (\pi^2\ \rho)^{(1/3)}},$$
$$C\_chi = 0.53 + 0.87\ \chi^2 + 0.5\ \chi^4 + 2.26\ \chi^6, tau\_w = \frac{norm\_drho^2}{8\ \rho},$$
$$C\_chi\_eps = \frac{C\_chi}{\left(1 + \frac{1}{2}\ eps^2 \left(\frac{1}{(1+\chi)^{(4/3)}} + \frac{1}{(1-\chi)^{(4/3)}}\right)\right)^4},$$
$$ma = \max(epsilon\_cGGA\_1\_0, epsilon\_cGGA), mb = \max(epsilon\_cGGA, epsilon\_cGGA\_0\_$$
$$epsilon\_cRevPKZB = epsilon\_cGGA$$
$$+ \frac{epsilon\_cGGA\ C\_chi\_eps\ tau\_w^2 - (1 + C\_chi\_eps)\ tau\_w^2 \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho}\right)}{\tau^2},$$
$$\left. ec = \rho\ epsilon\_cRevPKZB \left(1 + \frac{d\ epsilon\_cRevPKZB\ tau\_w^3}{\tau^3}\right) \right]$$

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

$$\{\pi, \rho, \tau, norm\_drho, epsilon\_cGGA\_1\_0, epsilon\_cGGA, norm\_drhoa, norm\_drhob,$$
$$epsilon\_cGGA\_0\_1, rhoa, rhob\}$$

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);
```

$$eqpbe1 := t = \frac{norm\_drho}{2\,\varphi\,k\_s\,\rho}$$

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

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

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

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

```
> 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));
```

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

```
> eqpbe6:=A=beta/gamma_var*(exp(-epsilon_c_unif/(gamma_var*phi^3*e^2/a_
  0))-1)^(-1);
```

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

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

$$eqpbe7 := epsilon\_cGGA = epsilon\_c\_unif + H$$

```
> eqpbe8:=beta=0.066725;
  eqpbe9:=gamma_var=(1-ln(2))/Pi^2;evalf(rhs(eqpbe9));
```

$$eqpbe8 := \beta = 0.066725$$

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

$$0.03109069086$$

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

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

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

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

$$\varphi = \frac{1}{2}(1+\chi)^{(2/3)} + \frac{1}{2}(1-\chi)^{(2/3)},\ k\_f = 3^{(1/3)}(\pi^2\rho)^{(1/3)},\ k\_s = 2\sqrt{\frac{k\_f}{\pi}},\ t = \frac{norm\_drho}{2\,\varphi\,k\_s\,\rho},$$

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

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

$$epsilon\_cGGA = epsilon\_c\_unif + H$$

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

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

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;
```

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

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

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

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

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

$$1.709920933$$

```
> G_uc:=-2*A*(1+alpha_1*rs)*ln(1+1/(2*A*(beta_1*rs^(1/2)+beta_2*rs+beta
  _3*rs^(3/2)+beta_4*rs^(p+1))));
```

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

```
> equc4:={p=1.0,A=0.031091,alpha_1=0.21370,beta_1=7.5957,beta_2=3.5876,
  beta_3=1.6382,
    beta_4=0.49294};
  equc5:=e_c_u_0=subs(equc4,G_uc);
```

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

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

```
> equc6:={p=1.0,A=0.015545,alpha_1=0.20548,beta_1=14.1189,beta_2=6.1977
  ,beta_3=3.3662,
    beta_4=0.62517};
  equc7:=e_c_u_1=subs(equc6,G_uc);
```

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

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

```
> equc8:={p=1.0,A=0.16887,alpha_1=0.11125,beta_1=10.357,beta_2=3.6231,b
  eta_3=0.88026,
    beta_4=0.49671};
  equc9:=alpha_c=-subs(equc8,G_uc);
```

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

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

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

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

$$f = \frac{(1+\chi)^{(4/3)} + (1-\chi)^{(4/3)} - 2}{2\,2^{(1/3)} - 2},$$

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

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

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

```
> 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)];
```

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

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

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

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

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

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

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

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

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

$$epsilon\_cGGA = epsilon\_c\_unif + H$$

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

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

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

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

```
> eqs_pbec3:=subs(map(x->x=x||_s1,loc(eqs_pbec2)),epsilon_cGGA_s1=epsil
  on_cGGA_1_0,
    rhob=0,norm_drho=norm_drhoa,norm_drhob=0,rho=rhoa,eqs_pbec2):
> unk(eqs_pbec3);
```

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

```
> eqs_pbec4:=subs(map(x->x=x||_s2,loc(eqs_pbec2)),epsilon_cGGA_s2=epsil
  on_cGGA_0_1,rhoa=0,norm_drho=norm_drhob,
    norm_drhoa=0,rho=rhob,eqs_pbec2):
> unk(eqs_pbec4);
```

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

```
> 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);
```

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

```
>
```

LDA

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

$$\{tau\_w\}$$

```
> 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=n
  orm_drho/2,
  [eqs_c2[i]$i=1..nops(eqs_c2),eqs_ex_lda_int[i]$i=1..nops(eqs_ex_lda_int
  ),energy=ec+ex_lda]):
> unk(eqs_lda);
```

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

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

$def\_lda := [chi\_s1, rs\_s1, e\_c\_u\_0\_s1, e\_c\_u\_1\_s1, alpha\_c\_s1, f\_ii\_0\_s1, f\_s1, epsilon\_c\_unif\_s1, beta\_s1, gamma\_var\_s1, phi\_s1, k\_f\_s1, k\_s\_s1, t\_s1, A\_s1, H\_s1, epsilon\_cGGA\_1\_0, chi\_s2, rs\_s2, e\_c\_u\_0\_s2, e\_c\_u\_1\_s2, alpha\_c\_s2, f\_ii\_0\_s2, f\_s2, epsilon\_c\_unif\_s2, beta\_s2, gamma\_var\_s2, phi\_s2, k\_f\_s2, k\_s\_s2, t\_s2, A\_s2, H\_s2, epsilon\_cGGA\_0\_1, \chi, rs, e\_c\_u\_0, e\_c\_u\_1, alpha\_c, f\_ii\_0, f, epsilon\_c\_unif, \beta, gamma\_var, \varphi, k\_f, k\_s, t, A, H, epsilon\_cGGA, d, norm\_dchi, eps, C\_chi, tau\_w, C\_chi\_eps, ma, mb, epsilon\_cRevPKZB, ec, \kappa, \mu, b, c, e\_var, p, z, \alpha, tildeq\_b, x, Fx, ex\_unif, ex\_lda, energy]$

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

$$ima := 58$$

$$imb := 59$$

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

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

$$eqMb := mb = \max(epsilon\_cGGA, epsilon\_cGGA\_0\_1)$$

```
> eqMas:=[ma=epsilon_cGGA,ma=epsilon_cGGA_1_0]:
  eqMbs:=[mb=epsilon_cGGA,mb=epsilon_cGGA_0_1]:
> arg_lda_names:=[rho,norm_drho,tau];
```

$$arg\_lda\_names := [\rho, norm\_drho, \tau]$$

```
> printlevel:=1;
```

$$printlevel := 1$$

```
> 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),ar
  g_lda_names):
   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);
```

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

```
> 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]);
```

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

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

$$-1$$

$$marho = epsilon\_cGGArho$$

$$mbrho = epsilon\_cGGA\_0\_1rho$$

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

$$corrMabEqs := [marho = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, epsilon\_cGGA\_1\_0rho, epsilon\_cGGArho), mbrho = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, epsilon\_cGGA\_0\_1rho, epsilon\_cGGArho), manorm\_drho = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, epsilon\_cGGA\_1\_0norm\_drho, epsilon\_cGGAnorm\_drho), mbnorm\_drho = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, epsilon\_cGGA\_0\_1norm\_drho, epsilon\_cGGAnorm\_drho)]$$

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

$$sostCorrMabEqs := [82 = (marho = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, epsilon\_cGGA\_1\_0rho, epsilon\_cGGArho)), 83 = (mbrho = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, epsilon\_cGGA\_0\_1rho, epsilon\_cGGArho)), 117 = (manorm\_drho = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, epsilon\_cGGA\_1\_0norm\_drho, epsilon\_cGGAnorm\_drho)), 118 = (mbnorm\_drho = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, epsilon\_cGGA\_0\_1norm\_drho epsilon\_cGGAnorm\_drho))]$$

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

```
> getDef(mbrho,eqs_lda3);
```

$$mbrho = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, epsilon\_cGGA\_0\_1rho, epsilon\_cGGArho)$$

```
> unk([op(eqs_lda3),result=deriv_rho]);
```

$$\{\pi, epsilon\_cGGA\_0\_1rho, \rho, \tau, norm\_drho, epsilon\_cGGA\_1\_0rho\}$$

```
> eqs_lda4:=enforceDependencies([my_tau=max(tau,tau_w),my_rho=rho,my_norm
  _drho=norm_drho,
   op(subs(tau=my_tau,rho=my_rho,norm_drho=my_norm_drho,eqs_lda3))]):
```

```
> res_eqs_lda:={energy,deriv_rho,deriv_norm_drho,deriv_tau};
  for my_symb in res_eqs_lda do
    print(my_symb,unk([op(eqs_lda4),result=my_symb]));
  end do;
```

$$res\_eqs\_lda := \{deriv\_norm\_drho, energy, deriv\_rho, deriv\_tau\}$$

$$deriv\_norm\_drho, \{\pi, \rho, \tau, norm\_drho\}$$

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

$$deriv\_rho, \{\pi, \rho, \tau, norm\_drho\}$$

$$deriv\_tau, \{\pi, \rho, \tau, norm\_drho\}$$

```
> glob_eqs_lda4:={my_rho,my_norm_drho,my_tau}union res_eqs_lda;
```

$$glob\_eqs\_lda4 := \{deriv\_norm\_drho, energy, deriv\_rho, deriv\_tau, my\_rho, my\_tau, my\_norm\_drho\}$$

```
> cs_eqs_lda4:=CompSeq(locals=loc(eqs_lda4)minus glob_eqs_lda4,
   globals=glob_eqs_lda4,params=[rho,norm_drho,tau],eqs_lda4):
  r_eqs_lda4:=convert(cs_eqs_lda4,procedure):
```

Fortran code

```
> Fortran(r_eqs_lda4,defaulttype=float,optimize);
```

```
Warning, the function names {myIF} are not recognized in the target
language
```

```
Warning, The following variable name replacements were made: ["cg",
"cg0", "cg1", "cg10", "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", "cg6", "cg7",
"cg8", "cg9"] = ["norm_drho", "alpharho", "t_s2norm_drho",
"tau_wnorm_drho", "rs_s2rho", "rs_s1", "t_s2rho", "epsilon_cGGArho",
"tildeq_btau", "tau_wrho", "alphatau", "epsilon_cGGA_0_1",
"epsilon_cRevPKZB", "epsilon_cGGA", "t_s1rho", "Hnorm_drho", "ex_unif",
"e_c_u_0rho", "tnorm_drho", "tildeq_bnorm_drho", "epsilon_cRevPKZBrho",
"pnorm_drho", "tildeq_b", "gamma_var", "e_c_u_1_s1", "k_f_s1",
"alphanorm_drho", "t_s1norm_drho", "znorm_drho", "epsilon_cGGA_1_0",
"k_s", "epsilon_cRevPKZBnorm_drho", "gamma_var_s2", "rs_s1rho",
"e_c_u_1_s2", "phi_s2", "e_c_u_1_s1rho", "e_c_u_1_s2rho", "tau_w",
"k_f_s1rho", "phi_s1", "k_s_s1", "t_s1", "A_s1", "rs_s2",
"mbnorm_drho", "k_s_s2", "A_s1rho", "manorm_drho", "gamma_var_s1",
"epsilon_cRevPKZBtau", "tildeq_brho", "r_eqs_lda4", "t_s2", "A_s2",
"e_c_u_0", "A_s2rho"]


      doubleprecision function cg55 (rho, cg, tau)
        doubleprecision deriv_norm_drho
        doubleprecision energy
        doubleprecision deriv_rho
        doubleprecision deriv_tau
        doubleprecision my_rho
        doubleprecision my_tau
        doubleprecision my_norm_drho
        common deriv_norm_drho, energy, deriv_rho, deriv_tau, my_rho,
my
     #_tau, my_norm_drho
        doubleprecision rho
        doubleprecision cg
        doubleprecision tau
        doubleprecision cg0
        doubleprecision cg1
        doubleprecision cg2
        doubleprecision cg3
        doubleprecision t136
        doubleprecision t247
        doubleprecision t259
        doubleprecision cg4
        doubleprecision cg5
        doubleprecision cg6
        doubleprecision cg7
        doubleprecision cg8
```

```
        doubleprecision t235
        doubleprecision t276
        doubleprecision t169
        doubleprecision t277
        doubleprecision t152
        doubleprecision t524
        doubleprecision t243
        doubleprecision t246
        doubleprecision t576
        doubleprecision t577
        doubleprecision t579
        doubleprecision t584
        doubleprecision t590
        doubleprecision t591
        doubleprecision cg9
        doubleprecision mbrho
        doubleprecision t599
        doubleprecision t38
        doubleprecision cg10
        doubleprecision t417
        doubleprecision t423
        doubleprecision cg11
        doubleprecision Fx
        doubleprecision cg12
        doubleprecision cg13
        doubleprecision t497
        doubleprecision t499
        doubleprecision t29
        doubleprecision t237
        doubleprecision t17
        doubleprecision t19
        doubleprecision cg14
        doubleprecision cg15
        doubleprecision t572
        doubleprecision t505
        doubleprecision t509
        doubleprecision t510
        doubleprecision t408
        doubleprecision t477
        doubleprecision t489
        integer t5
        doubleprecision t6
        doubleprecision t7
        doubleprecision t278
        doubleprecision t224
        doubleprecision cg16
        doubleprecision t67
```

```
      doubleprecision t68
      doubleprecision t197
      doubleprecision t434
      doubleprecision t85
      doubleprecision t86
      doubleprecision t217
      doubleprecision t219
      doubleprecision t220
      doubleprecision t223
      doubleprecision cg17
      doubleprecision t102
      doubleprecision t104
      doubleprecision t110
      doubleprecision prho
      doubleprecision zrho
      doubleprecision t172
      doubleprecision t173
      doubleprecision t25
      doubleprecision t114
      doubleprecision t119
      doubleprecision t122
      doubleprecision t123
      doubleprecision t227
      doubleprecision t228
      doubleprecision t230
      doubleprecision t231
      doubleprecision t232
      doubleprecision t236
      doubleprecision t282
      doubleprecision t586
      doubleprecision t390
      doubleprecision cg18
      doubleprecision z
      doubleprecision t14
      doubleprecision t367
      doubleprecision ma
      doubleprecision mb
      doubleprecision t210
      doubleprecision t369
      doubleprecision t303
      doubleprecision t305
      doubleprecision t306
      doubleprecision t307
      doubleprecision t309
      doubleprecision cg19
      doubleprecision cg20
      doubleprecision t787
```

```
      doubleprecision cg21
      doubleprecision cg22
      doubleprecision rsrho
      doubleprecision cg23
      doubleprecision t82
      doubleprecision cg24
      doubleprecision t556
      doubleprecision t564
      doubleprecision t566
      doubleprecision t78
      doubleprecision t79
      doubleprecision t512
      doubleprecision t517
      doubleprecision t468
      doubleprecision cg25
      doubleprecision cg26
      doubleprecision cg27
      doubleprecision t262
      doubleprecision t263
      doubleprecision t264
      doubleprecision t267
      doubleprecision t269
      doubleprecision t271
      doubleprecision p
      doubleprecision cg28
      doubleprecision cg29
      doubleprecision t283
      doubleprecision t284
      doubleprecision t285
      doubleprecision t288
      doubleprecision t290
      doubleprecision t293
      doubleprecision t296
      doubleprecision t145
      doubleprecision t147
      doubleprecision t148
      doubleprecision t149
      doubleprecision t151
      doubleprecision t155
      doubleprecision t248
      doubleprecision ztau
      integer t2
      integer t3
      integer t4
      doubleprecision cg30
      doubleprecision t435
      doubleprecision t437
```

```
doubleprecision t450
doubleprecision t453
doubleprecision t156
doubleprecision t157
doubleprecision trho
doubleprecision t325
doubleprecision t334
doubleprecision t74
doubleprecision t75
doubleprecision t76
doubleprecision cg31
doubleprecision cg32
doubleprecision t51
doubleprecision cg33
doubleprecision t139
doubleprecision t140
doubleprecision t141
doubleprecision t142
doubleprecision t250
doubleprecision t251
doubleprecision t252
doubleprecision t254
doubleprecision t257
doubleprecision t258
doubleprecision t260
doubleprecision t261
doubleprecision t622
logical t601
doubleprecision t604
doubleprecision t607
doubleprecision t380
doubleprecision t381
doubleprecision t239
doubleprecision t225
doubleprecision t226
doubleprecision cg34
doubleprecision cg35
integer t1
doubleprecision t176
doubleprecision cg36
doubleprecision cg37
doubleprecision cg38
doubleprecision Arho
doubleprecision t87
doubleprecision t374
doubleprecision t376
doubleprecision t66
```

```
        doubleprecision t9
        doubleprecision t674
        doubleprecision t676
        doubleprecision t163
        doubleprecision t164
        doubleprecision t165
        doubleprecision t168
        doubleprecision A
        doubleprecision cg39
        doubleprecision cg40
        doubleprecision cg41
        doubleprecision t759
        doubleprecision t213
        doubleprecision cg42
        doubleprecision alpha
        doubleprecision t342
        doubleprecision t343
        doubleprecision t348
        doubleprecision cg43
        doubleprecision t214
        doubleprecision t215
        doubleprecision cg44
        doubleprecision cg45
        doubleprecision cg46
        doubleprecision cg47
        doubleprecision cg48
        doubleprecision t158
        doubleprecision t159
        doubleprecision t160
        doubleprecision t161
        doubleprecision t315
        doubleprecision t316
        doubleprecision t319
        doubleprecision t320
        doubleprecision t92
        doubleprecision t95
        doubleprecision t96
        doubleprecision cg49
        doubleprecision t99
        doubleprecision t454
        integer t60
        integer t61
        doubleprecision t63
        doubleprecision rs
        doubleprecision t56
        doubleprecision t83
        doubleprecision t84
```

```
      doubleprecision t
      doubleprecision t401
      doubleprecision t69
      doubleprecision t72
      doubleprecision cg50
      doubleprecision marho
      doubleprecision t537
      doubleprecision t546
      doubleprecision t355
      doubleprecision t357
      doubleprecision t362
      doubleprecision t366
      doubleprecision t273
      doubleprecision cg51
      doubleprecision cg52
      doubleprecision t34
      doubleprecision t37
      doubleprecision t519
      doubleprecision t523
      doubleprecision t532
      logical t534
      doubleprecision cg53
      doubleprecision t58
      doubleprecision cg54
      doubleprecision t178
      doubleprecision t180
      doubleprecision t183
      doubleprecision t184
      doubleprecision t88
      doubleprecision t90
      doubleprecision t91
      my_rho = rho
      my_norm_drho = cg
      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)
      cg12 = dble(t4) * dble(t5) * t9 / 0.4D1
      t14 = sqrt(cg12)
      t17 = t14 * cg12
      t19 = cg12 ** 0.20D1
      t25 = log(0.1D1 + 0.1608182432D2 / (0.75957D1 * t14 + 0.35876D1
     #* cg12 + 0.16382D1 * t17 + 0.49294D0 * t19))
```

```
        t29 = 0.1D1 + 0.20548D0 * cg12
        t34 = 0.141189D2 * t14 + 0.61977D1 * cg12 + 0.33662D1 * t17 +
0.
     #62517D0 * t19
        t37 = 0.1D1 + 0.3216468318D2 / t34
        t38 = log(t37)
        cg3 = -0.31090D-1 * t29 * t38
        t51 = log(0.1D1 + 0.2960857464D1 / (0.10357D2 * t14 + 0.36231D1
     #* cg12 + 0.88026D0 * t17 + 0.49671D0 * t19))
        t56 = log(0.2D1)
        t58 = 0.3141592654D1 ** 2
        cg52 = (0.1D1 - t56) / t58
        t60 = t5 ** 2
        cg44 = dble(t60) / 0.2D1
        t61 = t1 * t60
        t63 = (t58 * rho) ** (0.1D1 / 0.3D1)
        cg30 = dble(t61) * t63 / 0.2D1
        t66 = sqrt(cg30 * t6)
        cg45 = 0.2D1 * t66
        t67 = 0.1D1 / cg44
        t68 = cg * t67
        t69 = 0.1D1 / cg45
        cg46 = t68 * t69 * t7 / 0.2D1
        t72 = 0.1D1 / cg52
        t74 = cg44 ** 2
        t75 = t74 * cg44
        t76 = 0.1D1 / t75
        t78 = exp(-cg3 * t72 * t76)
        t79 = t78 - 0.1D1
        cg47 = 0.66725D-1 * t72 / t79
        t82 = cg52 * t75
        t83 = cg46 ** 2
        t84 = t72 * t83
        t85 = cg47 * t83
        t86 = 0.1D1 + t85
        t87 = cg47 ** 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)
        cg34 = cg3 + t82 * t96
        cg48 = cg12
        t99 = sqrt(cg48)
        t102 = t99 * cg48
        t104 = cg48 ** 0.20D1
```

```
        t110 = log(0.1D1 + 0.1608182432D2 / (0.75957D1 * t99 +
0.35876D1
     # * cg48 + 0.16382D1 * t102 + 0.49294D0 * t104))
        t114 = 0.1D1 + 0.20548D0 * cg48
        t119 = 0.141189D2 * t99 + 0.61977D1 * cg48 + 0.33662D1 * t102 +
     #0.62517D0 * t104
        t122 = 0.1D1 + 0.3216468318D2 / t119
        t123 = log(t122)
        cg39 = -0.31090D-1 * t114 * t123
        t136 = log(0.1D1 + 0.2960857464D1 / (0.10357D2 * t99 +
0.36231D1
     # * cg48 + 0.88026D0 * t102 + 0.49671D0 * t104))
        cg37 = cg52
        cg4 = cg44
        t139 = sqrt(cg30 * t6)
        cg5 = 0.2D1 * t139
        t140 = 0.1D1 / cg4
        t141 = cg * t140
        t142 = 0.1D1 / cg5
        cg6 = t141 * t142 * t7 / 0.2D1
        t145 = 0.1D1 / cg37
        t147 = cg4 ** 2
        t148 = t147 * cg4
        t149 = 0.1D1 / t148
        t151 = exp(-cg39 * t145 * t149)
        t152 = t151 - 0.1D1
        cg7 = 0.66725D-1 * t145 / t152
        t155 = cg37 * t148
        t156 = cg6 ** 2
        t157 = t145 * t156
        t158 = cg7 * t156
        t159 = 0.1D1 + t158
        t160 = cg7 ** 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)
        cg18 = cg39 + t155 * t169
        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.
```

```
      #49294D0 * t178
         t183 = 0.1D1 + 0.1608182432D2 / t180
         t184 = log(t183)
         cg8 = -0.62182D-1 * t172 * t184
         t197 = log(0.1D1 + 0.3216468318D2 / (0.141189D2 * t173 +
0.61977
      #D1 * rs + 0.33662D1 * t176 + 0.62517D0 * t178))
         t210 = log(0.1D1 + 0.2960857464D1 / (0.10357D2 * t173 +
0.36231D
      #1 * rs + 0.88026D0 * t176 + 0.49671D0 * t178))
         cg29 = cg37
         t213 = sqrt(dble(t1) * t63 * t6)
         cg35 = 0.2D1 * t213
         t214 = 0.1D1 / cg35
         t215 = cg * t214
         t = t215 * t7 / 0.2D1
         t217 = 0.1D1 / cg29
         t219 = exp(-cg8 * 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)
         cg2 = cg8 + cg29 * t236
         t237 = cg ** 2
         cg42 = t237 * t7 / 0.8D1
         my_tau = max(tau, cg42)
         ma = max(cg34, cg2)
         mb = max(cg2, cg18)
         t239 = cg42 ** 2
         t243 = ma / 0.2D1 + mb / 0.2D1
         t246 = 0.53D0 * cg2 * t239 - 0.153D1 * t239 * t243
         t247 = my_tau ** 2
         t248 = 0.1D1 / t247
         cg19 = cg2 + t246 * t248
         t250 = rho * cg19
         t251 = t239 * cg42
         t252 = cg19 * t251
         t254 = 0.1D1 / t247 / my_tau
```

```
        t257 = 0.1D1 + 0.28D1 * t252 * t254
        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 / my_tau
        z = cg42 * 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
        cg28 = 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 = cg28 ** 2
        t293 = p ** 2
        t296 = sqrt(0.18D2 * t282 + 0.50D2 * t293)
        t303 = t288 * p + 0.146D3 / 0.2025D4 * t290 - 0.73D2 / 0.4050D4
     #* cg28 * t296 + 0.1895718785D-1 * t293 + 0.1102007148D0 * t282 +
0
     #.33738687D0 * t293 * p
        t305 = 0.1D1 + 0.1239758041D1 * p
        t306 = t305 ** 2
        t307 = 0.1D1 / t306
        t309 = 0.1D1 + 0.1243781095D1 * t303 * t307
        Fx = 0.1804D1 - 0.804D0 / t309
        cg22 = -0.3D1 / 0.4D1 * dble(t1) * t63 * t6
        t315 = rho * cg22
        energy = t250 * t257 + t315 * Fx
        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
        cg23 = -0.1328829340D-1 * rsrho * t184 + 0.9999999999D0 * t172
/
```

```
     # t325 * (0.3797850000D1 / t173 * rsrho + 0.35876D1 * rsrho +
0.245
     #7300000D1 * t173 * rsrho + 0.985880D0 * t334 * rsrho) / t183
        t342 = t63 ** 2
        t343 = 0.1D1 / t342
        t348 = cg35 ** 2
        trho = -cg / t348 * t7 / t213 * dble(t1) * t343 * t58 * t6 /
0.6
     #D1 - t215 * t319 / 0.2D1
        t355 = cg29 ** 2
        t357 = t220 ** 2
        Arho = 0.66725D-1 / t355 / t357 * cg23 * 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
        cg14 = cg23 + cg29 * (0.133450D0 * t362 * t232 * trho +
0.66725D
     #-1 * t224 * (t366 + t369) * t231 - 0.66725D-1 * t224 * t376 *
(t36
     #6 + t369 + 0.2D1 * A * t228 * Arho + 0.4D1 * t381 * trho)) * t390
        cg16 = -t237 * t319 / 0.8D1
        prho = -0.2D1 / 0.9D1 * t258 * t261 / t264 / t262 / rho
        zrho = cg16 * t269
        t401 = p / t282
        cg0 = 0.5D1 / 0.3D1 * prho * t271 - 0.5D1 / 0.3D1 * t401 * zrho
        t408 = t273 / t277 / t276
        cg54 = 0.9D1 / 0.20D2 * cg0 * t278 - 0.9D1 / 0.40D2 * t408 *
(0.
     #4D0 * cg0 * t273 + 0.4D0 * alpha * cg0) + 0.2D1 / 0.3D1 * prho
        t417 = z * t285
        t423 = t282 * z / t284 / t283
        t434 = cg28 / t296
        t435 = z * zrho
        t437 = p * prho
        t450 = t303 / t306 / t305
        t453 = t309 ** 2
        t454 = 0.1D1 / t453
        cg38 = -dble(t4) * dble(t5) * t320 / 0.12D2
        t468 = t34 ** 2
        t477 = cg12 ** 0.10D1
        cg40 = -0.638837320D-2 * cg38 * t38 + 0.1000000000D1 * t29 /
```

```
t46
     #8 * (0.7059450000D1 / t14 * cg38 + 0.61977D1 * cg38 +
0.5049300000
     #D1 * t14 * cg38 + 0.1250340D1 * t477 * cg38) / t37
        cg43 = dble(t61) * t343 * t58 / 0.6D1
        t489 = cg45 ** 2
        cg20 = -t68 / t489 * t7 / t66 * cg43 * t6 / 0.2D1 - t68 * t69 *
     #t319 / 0.2D1
        t497 = cg52 ** 2
        t499 = t79 ** 2
        cg50 = 0.66725D-1 / t497 / t499 * cg40 * t76 * t78
        t505 = t72 * cg46
        t509 = cg50 * t83
        t510 = cg47 * cg46
        t512 = 0.2D1 * t510 * cg20
        t517 = t90 ** 2
        t519 = t86 / t517
        t523 = t83 * cg46
        t524 = t87 * t523
        t532 = 0.1D1 / t95
        t534 = cg2 .lt. cg34
        marho = myIF(t534, cg40 + t82 * (0.133450D0 * t505 * t92 * cg20
     #+ 0.66725D-1 * t84 * (t509 + t512) * t91 - 0.66725D-1 * t84 *
t519
     # * (t509 + t512 + 0.2D1 * cg47 * t88 * cg50 + 0.4D1 * t524 *
cg20)
     #) * t532, cg14)
        cg11 = cg38
        t537 = t119 ** 2
        t546 = cg48 ** 0.10D1
        cg41 = -0.638837320D-2 * cg11 * t123 + 0.1000000000D1 * t114 /
t
     #537 * (0.7059450000D1 / t99 * cg11 + 0.61977D1 * cg11 +
0.50493000
     #00D1 * t99 * cg11 + 0.1250340D1 * t546 * cg11) / t122
        t556 = cg5 ** 2
        cg13 = -t141 / t556 * t7 / t139 * cg43 * t6 / 0.2D1 - t141 *
t14
     #2 * t319 / 0.2D1
        t564 = cg37 ** 2
        t566 = t152 ** 2
        cg9 = 0.66725D-1 / t564 / t566 * cg41 * t149 * t151
        t572 = t145 * cg6
        t576 = cg9 * t156
        t577 = cg6 * cg7
        t579 = 0.2D1 * t577 * cg13
        t584 = t163 ** 2
```

```
        t586 = t159 / t584
        t590 = t156 * cg6
        t591 = t160 * t590
        t599 = 0.1D1 / t168
        t601 = cg2 .lt. cg18
        mbrho = myIF(t601, cg41 + t155 * (0.133450D0 * t572 * t165 *
cg1
     #3 + 0.66725D-1 * t157 * (t576 + t579) * t164 - 0.66725D-1 * t157
*
     # t586 * (t576 + t579 + 0.2D1 * cg7 * t161 * cg9 + 0.4D1 * t591 *
c
     #g13)) * t599, cg14)
        t604 = cg2 * cg42
        t607 = cg42 * t243
        cg26 = cg14 + (0.53D0 * cg14 * t239 + 0.106D1 * t604 * cg16 -
0.
     #306D1 * t607 * cg16 - 0.153D1 * t239 * (marho / 0.2D1 + mbrho /
0.
     #2D1)) * t248
        t622 = cg19 * t239
        deriv_rho = cg19 * t257 + rho * cg26 * t257 + t250 * (0.28D1 *
c
     #g26 * t251 * t254 + 0.84D1 * t622 * t254 * cg16) + cg22 * Fx -
rho
     # * 0.3141592654D1 * dble(t1) * t343 * Fx / 0.4D1 + 0.1000000000D1
     #* t315 * t454 * (((0.318192D1 * t417 * zrho - 0.636384D1 * t423 *
     #zrho) * p + t288 * prho + 0.292D3 / 0.2025D4 * cg28 * cg54 -
0.73D
     #2 / 0.4050D4 * cg54 * t296 - 0.73D2 / 0.8100D4 * t434 * (0.36D2 *
     #t435 + 0.100D3 * t437) + 0.3791437570D-1 * t437 + 0.2204014296D0
*
     # t435 + 0.101216061D1 * t293 * prho) * t307 - 0.2479516082D1 *
t45
     #0 * prho)
        cg24 = t214 * t7 / 0.2D1
        cg21 = cg29 * (0.133450D0 * t362 * t232 * cg24 + 0.133450D0 *
t2
     #17 * t380 * A * cg24 * t231 - 0.66725D-1 * t224 * t376 * (0.2D1 *
     #t367 * cg24 + 0.4D1 * t381 * cg24)) * t390
        cg10 = cg * t7 / 0.4D1
        cg27 = cg * dble(t1) * t267 / 0.6D1
        cg33 = cg10 * t269
        cg31 = 0.5D1 / 0.3D1 * cg27 * t271 - 0.5D1 / 0.3D1 * t401 *
cg33
        cg25 = 0.9D1 / 0.20D2 * cg31 * t278 - 0.9D1 / 0.40D2 * t408 *
(0
     #.4D0 * cg31 * t273 + 0.4D0 * alpha * cg31) + 0.2D1 / 0.3D1 * cg27
```

```
        t674 = z * cg33
        t676 = p * cg27
        cg32 = t67 * t69 * t7 / 0.2D1
        cg51 = myIF(t534, t82 * (0.133450D0 * t505 * t92 * cg32 +
0.1334
     #50D0 * t72 * t523 * cg47 * cg32 * t91 - 0.66725D-1 * t84 * t519 *
     #(0.2D1 * t510 * cg32 + 0.4D1 * t524 * cg32)) * t532, cg21)
        cg1 = t140 * t142 * t7 / 0.2D1
        cg49 = myIF(t601, t155 * (0.133450D0 * t572 * t165 * cg1 +
0.133
     #450D0 * t145 * t590 * cg7 * cg1 * t164 - 0.66725D-1 * t157 * t586
     #* (0.2D1 * t577 * cg1 + 0.4D1 * t591 * cg1)) * t599, cg21)
        cg36 = cg21 + (0.53D0 * cg21 * t239 + 0.106D1 * t604 * cg10 -
0.
     #306D1 * t607 * cg10 - 0.153D1 * t239 * (cg51 / 0.2D1 + cg49 /
0.2D
     #1)) * t248
        deriv_norm_drho = rho * cg36 * t257 + t250 * (0.28D1 * cg36 *
t2
     #51 * t254 + 0.84D1 * t622 * t254 * cg10) + 0.1000000000D1 * t315
*
     # t454 * (((0.318192D1 * t417 * cg33 - 0.636384D1 * t423 * cg33) *
     #p + t288 * cg27 + 0.292D3 / 0.2025D4 * cg28 * cg25 - 0.73D2 /
0.40
     #50D4 * cg25 * t296 - 0.73D2 / 0.8100D4 * t434 * (0.36D2 * t674 +
0
     #.100D3 * t676) + 0.3791437570D-1 * t676 + 0.2204014296D0 * t674 +
     #0.101216061D1 * t293 * cg27) * t307 - 0.2479516082D1 * t450 *
cg27
     #)
        cg53 = -0.2D1 * t246 * t254
        t759 = t247 ** 2
        ztau = -cg42 * t248
        cg17 = -0.5D1 / 0.3D1 * t401 * ztau
        cg15 = 0.9D1 / 0.20D2 * cg17 * t278 - 0.9D1 / 0.40D2 * t408 *
(0
     #.4D0 * cg17 * t273 + 0.4D0 * alpha * cg17)
        t787 = z * ztau
        deriv_tau = rho * cg53 * t257 + t250 * (0.28D1 * cg53 * t251 *
t
     #254 - 0.84D1 * t252 / t759) + 0.1000000000D1 * t315 * t454 *
((0.3
     #18192D1 * t417 * ztau - 0.636384D1 * t423 * ztau) * p + 0.292D3 /
     #0.2025D4 * cg28 * cg15 - 0.73D2 / 0.4050D4 * cg15 * t296 - 0.73D2
     #/ 0.225D3 * t434 * t787 + 0.2204014296D0 * t787) * t307
        cg55 = deriv_tau
        return
```

end

Tests LDA

LSD

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

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

Tau -> tau_a correct? Calculate exchange separately (do ispin=1,2....)?

```
> eqs_ex_s1:=subs(op(map(x->x=x||_sp1,loc(eqs_ex_lda))),rho=2*rhoa,norm_d
  rho=2*norm_drhoa,tau=2*tau_a,
    eqs_ex_lda):
> unk(eqs_ex_s1);
```

$$\{\pi, norm\_drhoa, rhoa, tau\_a\}$$

```
> eqs_ex_s2:=subs(op(map(x->x=x||_sp2,loc(eqs_ex_lda))),rho=2*rhob,norm_d
  rho=2*norm_drhob,tau=2*tau_b,
    eqs_ex_lda):
> unk(eqs_ex_s2);
```

$$\{\pi, tau\_b, norm\_drhob, rhob\}$$

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

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

```
> eqs_lsd1:=[rho=rhoa+rhob,tau=tau_a+tau_b,op(eqs_ex_s1),op(eqs_ex_s2),op
  (eqs_c2),energy=ex_lda_sp1/2+ex_lda_sp2/2+ec]:
> unk(eqs_lsd1);
```

$$\{\pi, norm\_drho, tau\_b, norm\_drhoa, norm\_drhob, rhoa, rhob, tau\_a\}$$

```
> ima:=indiceDef(ma,eqs_lsd1);
  imb:=indiceDef(mb,eqs_lsd1);
```

$$ima := 88$$

$$imb := 89$$

```
> eqMa:=eqs_lsd1[ima];
  eqMb:=eqs_lsd1[imb];
```

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

$$eqMb := mb = \max(epsilon\_cGGA, epsilon\_cGGA\_0\_1)$$

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

```
> arg_lsd_names:=[rhoa,rhob,norm_drhoa,norm_drhob,norm_drho,tau_a,tau_b];
```

$$arg\_lsd\_names := [rhoa, rhob, norm\_drhoa, norm\_drhob, norm\_drho, tau\_a, tau\_b]$$

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

  deriv_lsd[i,j]:=calcDerivs(subsop(ima=eqMas[i],imb=eqMbs[j],eqs_lsd1),a
  rg_lsd_names):
   end do;
  end do;
  i:='i':j:='j':
> ims:= proc(eqs)local i,i1,i2;
   i1:=[indiceDef(ma,eqs),
     indiceDef(mb,eqs),
     op(map(x->indiceDef(x,eqs),[`||`(ma,arg_lsd_names[i])$i=1..5])),
     op(map(x->indiceDef(x,eqs),[`||`(mb,arg_lsd_names[i])$i=1..5]))];
   i2:=select(x->x>0,i1);
  end proc:
> eqss_lsd2:=[sostConst(eqs_lsd1),seq(seq(seq(deriv_lsd[i,j][ider],i=1..2
  ),j=1..2),
       ider=1..nops(arg_lsd_names))]:
> checkCompatible(eqss_lsd2,ims);
```

"def different for epsilon_cRevPKZBrhoa"

"incompatibility between" , 2, 4

"def different for epsilon_cRevPKZBrhoa"

"incompatibility between" , 2, 5

"def different for epsilon_cRevPKZBrhoa"

"incompatibility between" , 3, 4

"def different for epsilon_cRevPKZBrhoa"

"incompatibility between" , 3, 5

"def different for epsilon_cRevPKZBrhob"

"incompatibility between" , 6, 7

"def different for epsilon_cRevPKZBrhob"

"incompatibility between" , 6, 9

"def different for epsilon_cRevPKZBrhob"

"incompatibility between" , 7, 8

"def different for epsilon_cRevPKZBrhob"

"incompatibility between" , 8, 9

"def different for epsilon_cRevPKZBnorm_drhoa"

"incompatibility between" , 10, 11

"def different for epsilon_cRevPKZBnorm_drhoa"

"incompatibility between" , 10, 13

"def different for epsilon_cRevPKZBnorm_drhoa"

"incompatibility between" , 11, 12

"def different for epsilon_cRevPKZBnorm_drhoa"

"incompatibility between" , 12, 13

"def different for epsilon_cRevPKZBnorm_drhob"

"incompatibility between" , 14, 16

"def different for epsilon_cRevPKZBnorm_drhob"

"incompatibility between" , 14, 17

"def different for epsilon_cRevPKZBnorm_drhob"

"incompatibility between" , 15, 16

"def different for epsilon_cRevPKZBnorm_drhob"

"incompatibility between" , 15, 17

"def different for epsilon_cRevPKZBnorm_drho"

"incompatibility between" , 18, 19

"def different for epsilon_cRevPKZBnorm_drho"

"incompatibility between" , 18, 20

"def different for epsilon_cRevPKZBnorm_drho"

"incompatibility between" , 18, 21

"def different for epsilon_cRevPKZBnorm_drho"

"incompatibility between" , 19, 20

"def different for epsilon_cRevPKZBnorm_drho"

"incompatibility between" , 19, 21

"def different for epsilon_cRevPKZBnorm_drho"

"incompatibility between" , 20, 21

false

```
> myEq1:=getDef(epsilon_cRevPKZBrhoa,eqss_lsd2[2]);
```

$$myEq1 := epsilon\_cRevPKZBrhoa = epsilon\_cGGArhoa + \frac{1}{\tau^2}\left(epsilon\_cGGArhoa\ C\_chi\_eps\ tau\_w^2 \right.$$

$$+ epsilon\_cGGA\ C\_chi\_epsrhoa\ tau\_w^2 + 2\ epsilon\_cGGA\ C\_chi\_eps\ tau\_w\ tau\_wrhoa$$

$$- C\_chi\_epsrhoa\ tau\_w^2 \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right)$$

$$\left. - 2\ (1 + C\_chi\_eps)\ tau\_w \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right)\ tau\_wrhoa \right.$$

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

```
> op([2,2,1,6,4,5],myEq1);
```

$$\frac{rhob\ mbrhoa}{\rho}$$

```
> myEq2:=getDef(epsilon_cRevPKZBrhoa,eqss_lsd2[4]);
```

$$myEq2 := epsilon\_cRevPKZBrhoa = epsilon\_cGGArhoa + \frac{1}{\tau^{2}} \Bigg(epsilon\_cGGArhoa\ C\_chi\_eps\ tau\_w^{\,2}$$
$$+ epsilon\_cGGA\ C\_chi\_epsrhoa\ tau\_w^{\,2} + 2\ epsilon\_cGGA\ C\_chi\_eps\ tau\_w\ tau\_wrhoa$$
$$- C\_chi\_epsrhoa\ tau\_w^{\,2} \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right)$$
$$- 2\ (1 + C\_chi\_eps)\ tau\_w \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right)\ tau\_wrhoa$$
$$- (1 + C\_chi\_eps)\ tau\_w^{\,2} \left(\frac{ma}{\rho} - \frac{rhoa\ ma}{\rho^{2}} + \frac{rhoa\ marhoa}{\rho} - \frac{rhob\ mb}{\rho^{2}} \right) \Bigg)$$

```
> evalb(subsop([2,2,1,6,4,5]=0,myEq1)=myEq2);
```

$$true$$

```
>
```

```
> myEq1:=getDef(epsilon_cRevPKZBrhob,eqss_lsd2[6]);
```

$$myEq1 := epsilon\_cRevPKZBrhob = epsilon\_cGGArhob + \frac{1}{\tau^{2}} \Bigg(epsilon\_cGGArhob\ C\_chi\_eps\ tau\_w^{\,2}$$
$$+ epsilon\_cGGA\ C\_chi\_epsrhob\ tau\_w^{\,2} + 2\ epsilon\_cGGA\ C\_chi\_eps\ tau\_w\ tau\_wrhob$$
$$- C\_chi\_epsrhob\ tau\_w^{\,2} \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right)$$
$$- 2\ (1 + C\_chi\_eps)\ tau\_w \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right)\ tau\_wrhob$$
$$- (1 + C\_chi\_eps)\ tau\_w^{\,2} \left(- \frac{rhoa\ ma}{\rho^{2}} + \frac{rhoa\ marhob}{\rho} + \frac{mb}{\rho} - \frac{rhob\ mb}{\rho^{2}} + \frac{rhob\ mbrhob}{\rho} \right) \Bigg)$$

```
> op([2,2,1,6,4,2],myEq1);
```

$$\frac{rhoa\ marhob}{\rho}$$

```
> myEq2:=getDef(epsilon_cRevPKZBrhob,eqss_lsd2[7]);
```

$$myEq2 := epsilon\_cRevPKZBrhob = epsilon\_cGGArhob + \frac{1}{\tau^{2}} \Bigg(epsilon\_cGGArhob\ C\_chi\_eps\ tau\_w^{\,2}$$

$$+ epsilon\_cGGA\ C\_chi\_epsrhob\ tau\_w^{2} + 2\ epsilon\_cGGA\ C\_chi\_eps\ tau\_w\ tau\_wrhob$$
$$- C\_chi\_epsrhob\ tau\_w^{2}\left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho}\right)$$
$$- 2\ (1 + C\_chi\_eps)\ tau\_w\left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho}\right)\ tau\_wrhob$$
$$- (1 + C\_chi\_eps)\ tau\_w^{2}\left(-\frac{rhoa\ ma}{\rho^{2}} + \frac{mb}{\rho} - \frac{rhob\ mb}{\rho^{2}} + \frac{rhob\ mbrhob}{\rho}\right)\Bigg)$$

```
> evalb(subsop([2,2,1,6,4,2]=0,myEq1)=myEq2);
```

$$true$$

```
>
```

```
> myEq1:=getDef(epsilon_cRevPKZBnorm_drhoa,eqss_lsd2[10]);
```

$$myEq1 := epsilon\_cRevPKZBnorm\_drhoa = \frac{1}{\tau^{2}}\left(epsilon\_cGGA\ C\_chi\_epsnorm\_drhoa\ tau\_w^{2}\right.$$
$$\left. - C\_chi\_epsnorm\_drhoa\ tau\_w^{2}\left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho}\right)\right)$$

```
> myEq2:=getDef(epsilon_cRevPKZBnorm_drhoa,eqss_lsd2[11]);
```

$$myEq2 := epsilon\_cRevPKZBnorm\_drhoa = \frac{1}{\tau^{2}}\left(epsilon\_cGGA\ C\_chi\_epsnorm\_drhoa\ tau\_w^{2}\right.$$
$$- C\_chi\_epsnorm\_drhoa\ tau\_w^{2}\left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho}\right)$$
$$\left. - \frac{(1 + C\_chi\_eps)\ tau\_w^{2}\ rhoa\ manorm\_drhoa}{\rho}\right)$$

```
> op([2,1,3],myEq2);
```

$$- \frac{(1 + C\_chi\_eps)\ tau\_w^{2}\ rhoa\ manorm\_drhoa}{\rho}$$

```
> evalb(subsop([2,1,3]=0,myEq2)=myEq1);
```

$$true$$

```
>
```

```
> myEq1:=getDef(epsilon_cRevPKZBnorm_drhob,eqss_lsd2[14]);
```

$$myEq1 := epsilon\_cRevPKZBnorm\_drhob = \frac{1}{\tau^{2}}\left(epsilon\_cGGA\ C\_chi\_epsnorm\_drhob\ tau\_w^{2}\right.$$
$$\left. - C\_chi\_epsnorm\_drhob\ tau\_w^{2}\left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho}\right)\right)$$

```
> myEq2:=getDef(epsilon_cRevPKZBnorm_drhob,eqss_lsd2[16]);
```

$$myEq2 := epsilon\_cRevPKZBnorm\_drhob = \frac{1}{\tau^2}\left(epsilon\_cGGA\ C\_chi\_epsnorm\_drhob\ tau\_w^2 \right.$$
$$- C\_chi\_epsnorm\_drhob\ tau\_w^2 \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right)$$
$$\left. - \frac{(1 + C\_chi\_eps)\ tau\_w^2\ rhob\ mbnorm\_drhob}{\rho} \right)$$

```
> op([2,1,3],myEq2);
```

$$- \frac{(1 + C\_chi\_eps)\ tau\_w^2\ rhob\ mbnorm\_drhob}{\rho}$$

```
> evalb(subsop([2,1,3]=0,myEq2)=myEq1);
```

$$true$$

```
>
```

```
> myEq1:=getDef(epsilon_cRevPKZBnorm_drho,eqss_lsd2[18]);
```

$$myEq1 := epsilon\_cRevPKZBnorm\_drho = epsilon\_cGGAnorm\_drho + \frac{1}{\tau^2}\left(epsilon\_cGGAnorm\_drho \right.$$
$$C\_chi\_eps\ tau\_w^2 + epsilon\_cGGA\ C\_chi\_epsnorm\_drho\ tau\_w^2$$
$$+ 2\ epsilon\_cGGA\ C\_chi\_eps\ tau\_w\ tau\_wnorm\_drho$$
$$- C\_chi\_epsnorm\_drho\ tau\_w^2 \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right)$$
$$- 2\ (1 + C\_chi\_eps)\ tau\_w \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right) tau\_wnorm\_drho$$
$$\left. - (1 + C\_chi\_eps)\ tau\_w^2 \left(\frac{rhoa\ manorm\_drho}{\rho} + \frac{rhob\ mbnorm\_drho}{\rho} \right) \right)$$

```
> op([2,2,1,6,4,1],myEq1);
```

$$\frac{rhoa\ manorm\_drho}{\rho}$$

```
> myEq2:=getDef(epsilon_cRevPKZBnorm_drho,eqss_lsd2[19]);
```

$$myEq2 := epsilon\_cRevPKZBnorm\_drho = epsilon\_cGGAnorm\_drho + \frac{1}{\tau^2}\left(epsilon\_cGGAnorm\_drho \right.$$
$$C\_chi\_eps\ tau\_w^2 + epsilon\_cGGA\ C\_chi\_epsnorm\_drho\ tau\_w^2$$
$$+ 2\ epsilon\_cGGA\ C\_chi\_eps\ tau\_w\ tau\_wnorm\_drho$$
$$- C\_chi\_epsnorm\_drho\ tau\_w^2 \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right)$$
$$- 2\ (1 + C\_chi\_eps)\ tau\_w \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right) tau\_wnorm\_drho$$

$$- \frac{(1 + C\_chi\_eps)\ tau\_w^{2}\ rhob\ mbnorm\_drho}{\rho} \Bigg)$$

```
> evalb(subsop([2,2,1,6,4,1]=0,myEq1)=myEq2);
```

$$true$$

```
>
> eqss_lsd3:=subsop(10=op(11,eqss_lsd2),11=op(10,eqss_lsd2),14=op(16,eqss
  _lsd2),16=op(14,eqss_lsd2),
                    eqss_lsd2):
```

Order sequence defs

```
> def_eqss_lsd3:=map(definizioni,eqss_lsd3):
> allDefs_eqs_lsd3:=combineDefs(def_eqss_lsd3):
> eqs_lsd4:=combineEqs(allDefs_eqs_lsd3,eqss_lsd3,def_eqss_lsd3):
> unk(eqs_lsd4);
```

$$\{\pi, norm\_drho, tau\_b, norm\_drhoa, norm\_drhob, rhoa, rhob, tau\_a\}$$

```
> getDef(epsilon_cRevPKZBnorm_drhoa,eqs_lsd4);
  getDef(epsilon_cRevPKZBnorm_drhob,eqs_lsd4);
```

$$epsilon\_cRevPKZBnorm\_drhoa = \frac{1}{\tau^{2}} \Bigg(epsilon\_cGGA\ C\_chi\_epsnorm\_drhoa\ tau\_w^{2} - C\_chi\_epsnorm\_drhoa\ tau\_w^{2} \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right) - \frac{(1 + C\_chi\_eps)\ tau\_w^{2}\ rhoa\ manorm\_drhoa}{\rho} \Bigg)$$

$$epsilon\_cRevPKZBnorm\_drhob = \frac{1}{\tau^{2}} \Bigg(epsilon\_cGGA\ C\_chi\_epsnorm\_drhob\ tau\_w^{2} - C\_chi\_epsnorm\_drhob\ tau\_w^{2} \left(\frac{rhoa\ ma}{\rho} + \frac{rhob\ mb}{\rho} \right) - \frac{(1 + C\_chi\_eps)\ tau\_w^{2}\ rhob\ mbnorm\_drhob}{\rho} \Bigg)$$

```
> getDef(ma,eqs_lsd4);
  getDef(mb,eqs_lsd4);
  proc() local der;
   for der in [rhoa,rhob,norm_drhoa,norm_drhob,norm_drho] do
     print(getDef(ma||der,eqs_lsd4));
     print(getDef(mb||der,eqs_lsd4));
   end do;
```

```
  end proc();
```

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

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

$$marhoa = epsilon\_cGGArhoa$$

$$mbrhoa = epsilon\_cGGArhoa$$

$$marhob = epsilon\_cGGArhob$$

$$mbrhob = epsilon\_cGGArhob$$

$$manorm\_drhoa = epsilon\_cGGA\_1\_0norm\_drhoa$$

$$0$$

$$0$$

$$mbnorm\_drhob = epsilon\_cGGA\_0\_1norm\_drhob$$

$$manorm\_drho = epsilon\_cGGAnorm\_drho$$

$$mbnorm\_drho = epsilon\_cGGAnorm\_drho$$

```
> corrMabEqs:=proc() local res,der;
  res:=[];
  for der in [rhoa] do
    res:=[op(res),
      ma||der=myIF(epsilon_cGGA_1_0>epsilon_cGGA,
                   epsilon_cGGA_1_0||der,epsilon_cGGA||der),
      mb||der=myIF(epsilon_cGGA_0_1>epsilon_cGGA,
                   0,epsilon_cGGA||der)
  ];
  end do;
  for der in [rhob] do
    res:=[op(res),
      ma||der=myIF(epsilon_cGGA_1_0>epsilon_cGGA,
                   0,epsilon_cGGA||der),
      mb||der=myIF(epsilon_cGGA_0_1>epsilon_cGGA,
                   epsilon_cGGA_0_1||der,epsilon_cGGA||der)
  ];
  end do;
  for der in [norm_drhoa] do
    res:=[op(res),
      ma||der=myIF(epsilon_cGGA_1_0>epsilon_cGGA,
                   epsilon_cGGA_1_0||der,0)
  ];
  end do;
  for der in [norm_drhob] do
    res:=[op(res),
      mb||der=myIF(epsilon_cGGA_0_1>epsilon_cGGA,
                   epsilon_cGGA_0_1||der,0)
  ];
  end do;
```

```
  for der in [norm_drho] do
    res:=[op(res),
      ma||der=myIF(epsilon_cGGA_1_0>epsilon_cGGA,
                   0,epsilon_cGGA||der),
      mb||der=myIF(epsilon_cGGA_0_1>epsilon_cGGA,
                   0,epsilon_cGGA||der)
  ];
  end do;
  #subs(myIF=`if`,res);
  end proc();
```

$$corrMabEqs := [marhoa = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, epsilon\_cGGA\_1\_0rhoa, epsilon\_cGGArhoa), mbrhoa = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, 0, epsilon\_cGGArhoa), marhob = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, 0, epsilon\_cGGArhob), mbrhob = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, epsilon\_cGGA\_0\_1rhob, epsilon\_cGGArhob), manorm\_drhoa = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, epsilon\_cGGA\_1\_0norm\_drhoa, 0), mbnorm\_drhob = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, epsilon\_cGGA\_0\_1norm\_drhob, 0), manorm\_drho = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, 0, epsilon\_cGGAnorm\_drho), mbnorm\_drho = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, 0, epsilon\_cGGAnorm\_drho)]$$

```
> sostCorrMabEqs:=[seq(indiceDef(lhs(corrMabEqs[i]),eqs_lsd4)=(corrMabEqs
  [i]),i=1..nops(corrMabEqs))];
```

$$sostCorrMabEqs := [121 = (marhoa = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, epsilon\_cGGA\_1\_0rhoa, epsilon\_cGGArhoa)), 122 = (mbrhoa = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, 0, epsilon\_cGGArhoa)), 163 = (marhob = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, 0, epsilon\_cGGArhob)), 164 = (mbrhob = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, epsilon\_cGGA\_0\_1rhob, epsilon\_cGGArhob)), 191 = (manorm\_drhoa = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, epsilon\_cGGA\_1\_0norm\_drhoa, 0)), 209 = (mbnorm\_drhob = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, epsilon\_cGGA\_0\_1norm\_drhob, 0)), 220 = (manorm\_drho = myIF(epsilon\_cGGA < epsilon\_cGGA\_1\_0, 0, epsilon\_cGGAnorm\_drho)), 221 = (mbnorm\_drho = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, 0, epsilon\_cGGAnorm\_drho))]$$

```
> eqs_lsd5:=subsop(op(sostCorrMabEqs),eqs_lsd4):
```

```
> getDef(mbrhoa,eqs_lsd5);
```

$$mbrhoa = myIF(epsilon\_cGGA < epsilon\_cGGA\_0\_1, 0, epsilon\_cGGArhoa)$$

```
> unk([op(eqs_lsd5),result=deriv_rhoa]);
```

$$\{\pi, norm\_drho, tau\_b, norm\_drhoa, norm\_drhob, rhoa, rhob, epsilon\_cGGA\_1\_0rhoa, tau\_a\}$$

```
> eqs_lsd6:=enforceDependencies([my_tau_a=max(tau_a,norm_drhoa^2/(8*rhoa)
  ),

  my_tau_b=max(tau_b,norm_drhob^2/(8*rhob)),my_rhoa=rhoa,my_rhob=rhob,

  my_norm_drho=min(norm_drho,8*rho*(my_tau_a+my_tau_b)),my_norm_drhoa=nor
  m_drhoa,
      my_norm_drhob=norm_drhob,
      op(subs(tau_a=my_tau_a,tau_b=my_tau_b,rhoa=my_rhoa,rhob=my_rhob,
```

```
norm_drho=my_norm_drho,norm_drhoa=my_norm_drhoa,norm_drhob=my_norm_drho
b,
          eqs_lsd5))]):
```

```
> res_eqs_lsd:={energy,deriv_rhoa,deriv_rhob,deriv_norm_drhoa,deriv_norm_
drhob,deriv_norm_drho,
              deriv_tau_a,deriv_tau_b};
for my_symb in res_eqs_lsd do
  print(my_symb,unk([op(eqs_lsd6),result=my_symb])minus
convert(arg_lsd_names,set));
end do;
```

$$res\_eqs\_lsd := \{deriv\_norm\_drho, deriv\_rhoa, deriv\_rhob, deriv\_tau\_a, deriv\_tau\_b, energy, deriv\_norm\_drhoa, deriv\_norm\_drhob\}$$

$$deriv\_norm\_drho, \{\pi\}$$

$$deriv\_rhoa, \{\pi\}$$

$$deriv\_rhob, \{\pi\}$$

$$deriv\_tau\_a, \{\pi\}$$

$$deriv\_tau\_b, \{\pi\}$$

$$energy, \{\pi\}$$

$$deriv\_norm\_drhoa, \{\pi\}$$

$$deriv\_norm\_drhob, \{\pi\}$$

```
> glob_eqs_lsd6:={my_rhoa,my_rhob,my_norm_drho,my_norm_drhoa,my_norm_drho
b,my_tau_a,my_tau_b}union res_eqs_lsd;
```

$$glob\_eqs\_lsd6 := \{my\_tau\_a, deriv\_norm\_drho, deriv\_rhoa, deriv\_rhob, deriv\_tau\_a, deriv\_tau\_b, my\_norm\_drho, my\_rhoa, my\_rhob, energy, my\_tau\_b, deriv\_norm\_drhoa, deriv\_norm\_drhob, my\_norm\_drhoa, my\_norm\_drhob\}$$

```
> cs_eqs_lsd6:=CompSeq(locals=loc(eqs_lsd6)minus glob_eqs_lsd6,

globals=glob_eqs_lsd6,params=[rhoa,rhob,norm_drhoa,norm_drhob,norm_drho
,tau_a,tau_b],eqs_lsd6):
r_eqs_lsd6:=convert(cs_eqs_lsd6,procedure):
```

Fortran code

```
> Fortran(r_eqs_lsd6,defaulttype=float,optimize);
```

```
Warning, the function names {myIF} 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", "cg108", "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"] = ["norm_drhoa", "norm_drhob", "norm_drho",
"manorm_drho", "tildeq_b_sp2norm_drhob", "t_s2norm_drhob",
"mbnorm_drhob", "tau_wnorm_drho", "alpha_sp1tau_a",
"epsilon_cRevPKZBtau_a", "epsilon_cRevPKZBtau_b", "alpha_sp2tau_b",
"r_eqs_lsd6", "tnorm_drho", "rs_s1", "Fx_sp2", "Hnorm_drho",
"e_c_u_0rhoa", "tildeq_b_sp1rhoa", "epsilon_c_unifrhoa",
"epsilon_cGGArhoa", "tildeq_b_sp2rhob", "tau_a", "z_sp2rhob",
"epsilon_cGGA_1_0", "epsilon_cGGA_0_1", "epsilon_cRevPKZB",
"p_sp1rhoa", "epsilon_cRevPKZBnorm_drhoa",
"epsilon_cRevPKZBnorm_drhob", "epsilon_cRevPKZBnorm_drho",
"mbnorm_drho", "p_sp1", "tau_b", "z_sp1", "Fx_sp1", "p_sp2", "z_sp2",
"tau_w_sp1", "alpha_sp1", "ex_unif_sp1", "tau_w_sp2", "alpha_sp2",
"ex_unif_sp2", "gamma_var", "e_c_u_1_s1", "e_c_u_1_s2", "C_chi_eps",
"z_sp1rhoa", "tau_wrhoa", "t_s1rhoa", "A_s1rhoa", "p_sp2rhob",
"e_c_u_0rhob", "rs_s2rhob", "t_s1norm_drhoa", "t_s2rhob", "A_s2rhob",
"z_sp1tau_a", "z_sp2tau_b", "tildeq_b_sp1", "e_c_u_1_s2rhob",
"tildeq_b_sp2", "gamma_var_s1", "gamma_var_s2", "alpha_sp1rhoa",
"C_chi_epsrhob", "epsilon_c_unif", "phi_s1", "k_s_s1", "t_s1", "A_s1",
"rs_s2", "phi_s2", "k_s_s2", "t_s2", "e_c_u_0", "A_s2", "alpha_c",
"k_s", "epsilon_cGGA", "C_chi", "tau_w", "chirhoa", "phirhoa",
"k_frhoa", "C_chi_epsrhoa", "e_c_u_1_s1rhoa", "tau_wrhob",
"alpha_sp2rhob", "epsilon_cRevPKZBrhoa", "chirhob",
"epsilon_cRevPKZBrhob", "epsilon_c_unifrhob", "epsilon_cGGArhob",
"alpha_sp1norm_drhoa", "C_chi_epsnorm_drhoa", "alpha_sp2norm_drhob",
"C_chi_epsnorm_drhob", "rs_s1rhoa", "C_chi_epsnorm_drho",
"tildeq_b_sp1tau_a", "tildeq_b_sp2tau_b", "phirhob", "p_sp1norm_drhoa",
"z_sp1norm_drhoa", "manorm_drhoa", "p_sp2norm_drhob",
"z_sp2norm_drhob", "tildeq_b_sp1norm_drhoa"]


      doubleprecision function cg108 (rhoa, rhob, cg, cg0, cg1, cg2,
cg3
     #)
        doubleprecision my_tau_a
        doubleprecision deriv_norm_drho
        doubleprecision deriv_rhoa
        doubleprecision deriv_rhob
        doubleprecision deriv_tau_a
        doubleprecision deriv_tau_b
        doubleprecision my_norm_drho
        doubleprecision my_rhoa
```

```
        doubleprecision my_rhob
        doubleprecision energy
        doubleprecision my_tau_b
        doubleprecision deriv_norm_drhoa
        doubleprecision deriv_norm_drhob
        doubleprecision my_norm_drhoa
        doubleprecision my_norm_drhob
        common my_tau_a, deriv_norm_drho, deriv_rhoa, deriv_rhob,
deriv_
     #tau_a, deriv_tau_b, my_norm_drho, my_rhoa, my_rhob, energy,
my_tau
     #_b, deriv_norm_drhoa, deriv_norm_drhob, my_norm_drhoa,
my_norm_drh
     #ob
        doubleprecision rhoa
        doubleprecision rhob
        doubleprecision cg
        doubleprecision cg0
        doubleprecision cg1
        doubleprecision cg2
        doubleprecision cg3
        doubleprecision t318
        doubleprecision t126
        doubleprecision t52
        integer t141
        doubleprecision t368
        doubleprecision t286
        doubleprecision t279
        doubleprecision t718
        doubleprecision t422
        doubleprecision cg4
        doubleprecision t206
        doubleprecision cg5
        doubleprecision t348
        doubleprecision t103
        doubleprecision t86
        doubleprecision t378
        doubleprecision t352
        doubleprecision t230
        doubleprecision t1089
        doubleprecision t370
        doubleprecision rsrhoa
        doubleprecision t627
        doubleprecision t351
        doubleprecision t610
        doubleprecision t686
        doubleprecision cg6
```

```
      doubleprecision t391
      doubleprecision t202
      logical t806
      doubleprecision t129
      doubleprecision t371
      doubleprecision t213
      doubleprecision t290
      doubleprecision t631
      doubleprecision t682
      doubleprecision t328
      doubleprecision t220
      doubleprecision t665
      doubleprecision t639
      doubleprecision t349
      doubleprecision t280
      doubleprecision cg7
      doubleprecision cg8
      doubleprecision t1047
      doubleprecision cg9
      doubleprecision t78
      doubleprecision cg10
      doubleprecision t395
      doubleprecision t282
      doubleprecision t283
      doubleprecision t1084
      doubleprecision t856
      doubleprecision t1091
      doubleprecision t563
      doubleprecision t546
      doubleprecision cg11
      doubleprecision t454
      doubleprecision t663
      doubleprecision t769
      doubleprecision t771
      doubleprecision cg12
      doubleprecision t374
      doubleprecision t387
      doubleprecision t409
      doubleprecision mbrhoa
      doubleprecision t158
      doubleprecision t571
      doubleprecision t299
      doubleprecision cg13
      doubleprecision t622
      doubleprecision t96
      doubleprecision t233
      doubleprecision t55
```

```
doubleprecision t443
doubleprecision t444
doubleprecision t445
doubleprecision t890
doubleprecision t889
doubleprecision t535
doubleprecision t577
doubleprecision t579
doubleprecision t970
doubleprecision t554
doubleprecision t184
doubleprecision t431
doubleprecision t670
doubleprecision t585
doubleprecision t355
doubleprecision t278
doubleprecision t526
doubleprecision cg14
doubleprecision t451
doubleprecision t382
doubleprecision cg15
doubleprecision t2
integer t12
doubleprecision t997
doubleprecision t878
doubleprecision cg16
doubleprecision cg17
doubleprecision cg18
doubleprecision cg19
doubleprecision t36
doubleprecision t241
doubleprecision t1418
logical t731
doubleprecision cg20
doubleprecision t863
doubleprecision t367
doubleprecision cg21
doubleprecision cg22
doubleprecision cg23
integer t140
doubleprecision t678
doubleprecision t403
doubleprecision cg24
doubleprecision cg25
doubleprecision cg26
doubleprecision cg27
doubleprecision t658
```

```
        doubleprecision t476
        doubleprecision t518
        doubleprecision t49
        doubleprecision cg28
        integer t1
        integer t139
        doubleprecision t1447
        doubleprecision t607
        doubleprecision frhoa
        doubleprecision t1056
        doubleprecision t357
        doubleprecision t469
        doubleprecision t485
        doubleprecision t701
        doubleprecision t795
        doubleprecision t197
        doubleprecision t619
        doubleprecision cg29
        doubleprecision cg30
        doubleprecision cg31
        doubleprecision cg32
        doubleprecision cg33
        doubleprecision cg34
        doubleprecision cg35
        doubleprecision cg36
        doubleprecision cg37
        doubleprecision cg38
        doubleprecision cg39
        doubleprecision cg40
        doubleprecision cg41
        doubleprecision cg42
        doubleprecision cg43
        doubleprecision cg44
        doubleprecision cg45
        doubleprecision cg46
        doubleprecision cg47
        doubleprecision cg48
        doubleprecision cg49
        doubleprecision cg50
        doubleprecision cg51
        doubleprecision cg52
        doubleprecision cg53
        doubleprecision cg54
        doubleprecision cg55
        doubleprecision cg56
        doubleprecision rho
        doubleprecision tau
```

```
      doubleprecision cg57
      doubleprecision cg58
      doubleprecision cg59
      doubleprecision cg60
      doubleprecision cg61
      doubleprecision cg62
      doubleprecision cg63
      doubleprecision cg64
      doubleprecision cg65
      doubleprecision cg66
      doubleprecision cg67
      doubleprecision cg68
      doubleprecision cg69
      doubleprecision cg70
      doubleprecision chi
      doubleprecision rs
      doubleprecision t388
      doubleprecision f
      doubleprecision phi
      doubleprecision cg71
      doubleprecision t
      doubleprecision A
      doubleprecision cg72
      doubleprecision eps
      doubleprecision cg73
      doubleprecision cg74
      doubleprecision cg75
      doubleprecision ma
      doubleprecision mb
      doubleprecision t414
      doubleprecision cg76
      doubleprecision cg77
      doubleprecision trhoa
      doubleprecision Arhoa
      doubleprecision cg78
      doubleprecision cg79
      doubleprecision cg80
      doubleprecision cg81
      doubleprecision marhoa
      doubleprecision cg82
      doubleprecision rsrhob
      doubleprecision frhob
      doubleprecision cg83
      doubleprecision cg84
      doubleprecision cg85
      doubleprecision cg86
      doubleprecision cg87
```

```
doubleprecision cg88
doubleprecision cg89
doubleprecision cg90
doubleprecision cg91
doubleprecision cg92
doubleprecision trhob
doubleprecision Arhob
doubleprecision cg93
doubleprecision t291
doubleprecision marhob
doubleprecision mbrhob
doubleprecision cg94
doubleprecision cg95
doubleprecision cg96
doubleprecision cg97
doubleprecision cg98
doubleprecision cg99
doubleprecision cg100
doubleprecision t804
doubleprecision cg101
doubleprecision cg102
doubleprecision cg103
doubleprecision cg104
doubleprecision t669
doubleprecision t441
doubleprecision cg105
doubleprecision cg106
doubleprecision cg107
doubleprecision t705
doubleprecision t713
doubleprecision t1401
doubleprecision t1096
doubleprecision t1068
doubleprecision t908
doubleprecision t909
doubleprecision t655
doubleprecision t3
integer t5
doubleprecision t6
doubleprecision t7
doubleprecision t9
integer t13
doubleprecision t14
doubleprecision t15
doubleprecision t16
doubleprecision t837
integer t18
```

```
        doubleprecision t19
        doubleprecision t20
        doubleprecision t21
        doubleprecision t22
        doubleprecision t25
        doubleprecision t27
        doubleprecision t30
        doubleprecision t32
        doubleprecision t35
        doubleprecision t37
        doubleprecision t41
        doubleprecision t42
        doubleprecision t43
        doubleprecision t44
        doubleprecision t47
        doubleprecision t62
        doubleprecision t64
        doubleprecision t65
        doubleprecision t66
        doubleprecision t68
        doubleprecision t71
        doubleprecision t72
        doubleprecision t796
        doubleprecision t74
        doubleprecision t928
        integer t80
        doubleprecision t81
        doubleprecision t82
        doubleprecision t83
        doubleprecision t102
        doubleprecision t104
        doubleprecision t105
        doubleprecision t88
        doubleprecision t91
        doubleprecision t93
        doubleprecision t97
        doubleprecision t98
        doubleprecision t1171
        doubleprecision t1078
        doubleprecision t108
        doubleprecision t110
        doubleprecision t113
        doubleprecision t116
        doubleprecision t123
        doubleprecision t125
        doubleprecision t127
        doubleprecision t133
```

```
doubleprecision t137
doubleprecision t143
doubleprecision t147
doubleprecision t150
doubleprecision t152
doubleprecision t557
doubleprecision t162
doubleprecision t167
doubleprecision t170
doubleprecision t171
doubleprecision t1415
integer t188
doubleprecision t189
integer t192
doubleprecision t194
doubleprecision t195
doubleprecision t196
doubleprecision t200
doubleprecision t203
doubleprecision t204
doubleprecision t207
doubleprecision t210
doubleprecision t211
doubleprecision t212
doubleprecision t214
doubleprecision t215
doubleprecision t216
doubleprecision t218
doubleprecision t219
doubleprecision t223
doubleprecision t224
doubleprecision t226
doubleprecision t235
doubleprecision t245
doubleprecision t250
doubleprecision t253
doubleprecision t254
doubleprecision t267
doubleprecision t270
doubleprecision t271
doubleprecision t272
doubleprecision t273
doubleprecision t276
doubleprecision t999
doubleprecision t1103
doubleprecision t287
doubleprecision t288
```

```
        doubleprecision t289
        doubleprecision t292
        doubleprecision t294
        doubleprecision t295
        doubleprecision t296
        doubleprecision t300
        doubleprecision t301
        doubleprecision t302
        doubleprecision t304
        doubleprecision t307
        doubleprecision t308
        doubleprecision t311
        doubleprecision t313
        doubleprecision t315
        doubleprecision t319
        doubleprecision t323
        doubleprecision t331
        doubleprecision t332
        doubleprecision t336
        doubleprecision t341
        doubleprecision t344
        doubleprecision t345
        doubleprecision t347
        doubleprecision t350
        doubleprecision t354
        doubleprecision t356
        doubleprecision t359
        doubleprecision t361
        doubleprecision t362
        doubleprecision t364
        doubleprecision t365
        doubleprecision t372
        doubleprecision t373
        doubleprecision t376
        doubleprecision t377
        doubleprecision t379
        doubleprecision t380
        doubleprecision t383
        doubleprecision t386
        doubleprecision t389
        doubleprecision t390
        doubleprecision t392
        doubleprecision t394
        doubleprecision t396
        doubleprecision t399
        doubleprecision t400
        doubleprecision t404
```

```
        doubleprecision t410
        doubleprecision t411
        integer t413
        doubleprecision t415
        doubleprecision t425
        doubleprecision t428
        doubleprecision t429
        doubleprecision t430
        doubleprecision t432
        doubleprecision t433
        doubleprecision t435
        doubleprecision t436
        doubleprecision t437
        doubleprecision t439
        doubleprecision t447
        doubleprecision t448
        doubleprecision t449
        doubleprecision t463
        doubleprecision t491
        doubleprecision t502
        doubleprecision t503
        doubleprecision t505
        doubleprecision t521
        doubleprecision t522
        doubleprecision t524
        doubleprecision t525
        doubleprecision t536
        doubleprecision t544
        doubleprecision t547
        doubleprecision t548
        doubleprecision t551
        doubleprecision t553
        doubleprecision t565
        doubleprecision t597
        doubleprecision t598
        doubleprecision t612
        doubleprecision t615
        doubleprecision t625
        doubleprecision t633
        doubleprecision t635
        doubleprecision t638
        doubleprecision t647
        doubleprecision t651
        doubleprecision t656
        doubleprecision t666
        doubleprecision t683
        doubleprecision t693
```

```
        doubleprecision t714
        doubleprecision t716
        doubleprecision t720
        doubleprecision t722
        doubleprecision t732
        doubleprecision t740
        doubleprecision t749
        doubleprecision t761
        doubleprecision t777
        doubleprecision t781
        doubleprecision t782
        doubleprecision t784
        doubleprecision t789
        doubleprecision t791
        doubleprecision t816
        doubleprecision t822
        doubleprecision t825
        doubleprecision t831
        doubleprecision t850
        doubleprecision t872
        doubleprecision t892
        doubleprecision t905
        doubleprecision t911
        doubleprecision t912
        doubleprecision t923
        doubleprecision t926
        doubleprecision t1039
        doubleprecision t1076
        doubleprecision t1088
        doubleprecision t1098
        doubleprecision t1102
        doubleprecision t1111
        doubleprecision t1173
        doubleprecision t1259
        doubleprecision t1261
        doubleprecision t1376
        doubleprecision t1422
        t1 = int(cg ** 2)
        t2 = 0.1D1 / rhoa
        t3 = dble(t1) * t2
        my_tau_a = max(t3 / 0.8D1, cg2)
        t5 = int(cg0 ** 2)
        t6 = 0.1D1 / rhob
        t7 = dble(t5) * t6
        my_tau_b = max(cg3, t7 / 0.8D1)
        my_rhoa = rhoa
        my_rhob = rhob
```

```
        my_norm_drhoa = cg
        my_norm_drhob = cg0
        rho = rhoa + rhob
        t9 = my_tau_a + my_tau_b
        my_norm_drho = min(cg1, 0.8D1 * rho * t9)
        tau = t9
        t12 = 3 ** (0.1D1 / 0.3D1)
        t13 = t1 * t12
        t14 = 0.3141592654D1 ** 2
        t15 = t14 ** (0.1D1 / 0.3D1)
        t16 = t15 ** 2
        t18 = 2 ** (0.1D1 / 0.3D1)
        t19 = 0.1D1 / t16 * dble(t18)
        t20 = rhoa ** 2
        t21 = rhoa ** (0.1D1 / 0.3D1)
        t22 = t21 ** 2
        t25 = t19 / t22 / t20
        cg29 = dble(t13) * t25 / 0.24D2
        cg34 = t3 / 0.4D1
        t27 = 0.1D1 / my_tau_a
        cg30 = cg34 * t27 / 0.2D1
        t30 = 0.1D1 / cg30 - 0.1D1
        cg35 = 0.5D1 / 0.3D1 * cg29 * t30
        t32 = cg35 - 0.1D1
        t35 = 0.1D1 + 0.4D0 * cg35 * t32
        t36 = sqrt(t35)
        t37 = 0.1D1 / t36
        cg54 = 0.9D1 / 0.20D2 * t32 * t37 + 0.2D1 / 0.3D1 * cg29
        t41 = cg30 ** 2
        t42 = 0.1D1 + t41
        t43 = t42 ** 2
        t44 = 0.1D1 / t43
        t47 = 0.10D2 / 0.81D2 + 0.159096D1 * t41 * t44
        t49 = cg54 ** 2
        t52 = cg29 ** 2
        t55 = sqrt(0.18D2 * t41 + 0.50D2 * t52)
        t62 = t47 * cg29 + 0.146D3 / 0.2025D4 * t49 - 0.73D2 / 0.4050D4
     #* cg54 * t55 + 0.1895718785D-1 * t52 + 0.1102007148D0 * t41 +
0.33
     #738687D0 * t52 * cg29
        t64 = 0.1D1 + 0.1239758041D1 * cg29
        t65 = t64 ** 2
        t66 = 0.1D1 / t65
        t68 = 0.1D1 + 0.1243781095D1 * t62 * t66
        cg31 = 0.1804D1 - 0.804D0 / t68
        t71 = 0.1D1 / 0.3141592654D1
        t72 = t71 * dble(t12)
```

```
        t74 = (t14 * rhoa) ** (0.1D1 / 0.3D1)
        cg36 = -0.3D1 / 0.4D1 * t72 * dble(t18) * t74
        t78 = rhoa * cg36
        t80 = t5 * t12
        t81 = rhob ** 2
        t82 = rhob ** (0.1D1 / 0.3D1)
        t83 = t82 ** 2
        t86 = t19 / t83 / t81
        cg32 = dble(t80) * t86 / 0.24D2
        cg37 = t7 / 0.4D1
        t88 = 0.1D1 / my_tau_b
        cg33 = cg37 * t88 / 0.2D1
        t91 = 0.1D1 / cg33 - 0.1D1
        cg38 = 0.5D1 / 0.3D1 * cg32 * t91
        t93 = cg38 - 0.1D1
        t96 = 0.1D1 + 0.4D0 * cg38 * t93
        t97 = sqrt(t96)
        t98 = 0.1D1 / t97
        cg56 = 0.9D1 / 0.20D2 * t93 * t98 + 0.2D1 / 0.3D1 * cg32
        t102 = cg33 ** 2
        t103 = 0.1D1 + t102
        t104 = t103 ** 2
        t105 = 0.1D1 / t104
        t108 = 0.10D2 / 0.81D2 + 0.159096D1 * t102 * t105
        t110 = cg56 ** 2
        t113 = cg32 ** 2
        t116 = sqrt(0.18D2 * t102 + 0.50D2 * t113)
        t123 = t108 * cg32 + 0.146D3 / 0.2025D4 * t110 - 0.73D2 /
0.4050
     #D4 * cg56 * t116 + 0.1895718785D-1 * t113 + 0.1102007148D0 * t102
     #+ 0.33738687D0 * t113 * cg32
        t125 = 0.1D1 + 0.1239758041D1 * cg32
        t126 = t125 ** 2
        t127 = 0.1D1 / t126
        t129 = 0.1D1 + 0.1243781095D1 * t123 * t127
        cg13 = 0.1804D1 - 0.804D0 / t129
        t133 = (t14 * rhob) ** (0.1D1 / 0.3D1)
        cg39 = -0.3D1 / 0.4D1 * t72 * dble(t18) * t133
        t137 = rhob * cg39
        t139 = 4 ** (0.1D1 / 0.3D1)
        t140 = t139 ** 2
        t141 = t12 * t140
        t143 = (t71 * t2) ** (0.1D1 / 0.3D1)
        cg12 = dble(t141) * t143 / 0.4D1
        t147 = sqrt(cg12)
        t150 = t147 * cg12
        t152 = cg12 ** 0.20D1
```

```
        t158 = log(0.1D1 + 0.1608182432D2 / (0.75957D1 * t147 +
0.35876D
     #1 * cg12 + 0.16382D1 * t150 + 0.49294D0 * t152))
        t162 = 0.1D1 + 0.20548D0 * cg12
        t167 = 0.141189D2 * t147 + 0.61977D1 * cg12 + 0.33662D1 * t150
+
     # 0.62517D0 * t152
        t170 = 0.1D1 + 0.3216468318D2 / t167
        t171 = log(t170)
        cg40 = -0.31090D-1 * t162 * t171

        t184 = log(0.1D1 + 0.2960857464D1 / (0.10357D2 * t147 +
0.36231D
     #1 * cg12 + 0.88026D0 * t150 + 0.49671D0 * t152))
        t188 = 1 / (2 * t18 - 2)
        t189 = log(0.2D1)
        cg57 = (0.1D1 - t189) / t14
        t192 = t18 ** 2
        cg61 = dble(t192) / 0.2D1
        t194 = sqrt(dble(t12) * t74 * t71)
        cg62 = 0.2D1 * t194
        t195 = 0.1D1 / cg61
        t196 = cg * t195
        t197 = 0.1D1 / cg62
        cg63 = t196 * t197 * t2 / 0.2D1
        t200 = 0.1D1 / cg57
        t202 = cg61 ** 2
        t203 = t202 * cg61
        t204 = 0.1D1 / t203
        t206 = exp(-cg40 * t200 * t204)
        t207 = t206 - 0.1D1
        cg64 = 0.66725D-1 * t200 / t207
        t210 = cg57 * t203
        t211 = cg63 ** 2
        t212 = t200 * t211
        t213 = cg64 * t211
        t214 = 0.1D1 + t213
        t215 = cg64 ** 2
        t216 = t211 ** 2
        t218 = 0.1D1 + t213 + t215 * t216
        t219 = 0.1D1 / t218
        t220 = t214 * t219
        t223 = 0.1D1 + 0.66725D-1 * t212 * t220
        t224 = log(t223)
        cg21 = cg40 + t210 * t224
        t226 = (t71 * t6) ** (0.1D1 / 0.3D1)
        cg65 = dble(t141) * t226 / 0.4D1
```

```
        t230 = sqrt(cg65)
        t233 = t230 * cg65
        t235 = cg65 ** 0.20D1
        t241 = log(0.1D1 + 0.1608182432D2 / (0.75957D1 * t230 +
0.35876D
     #1 * cg65 + 0.16382D1 * t233 + 0.49294D0 * t235))
        t245 = 0.1D1 + 0.20548D0 * cg65
        t250 = 0.141189D2 * t230 + 0.61977D1 * cg65 + 0.33662D1 * t233
+
     # 0.62517D0 * t235
        t253 = 0.1D1 + 0.3216468318D2 / t250
        t254 = log(t253)
        cg41 = -0.31090D-1 * t245 * t254
        t267 = log(0.1D1 + 0.2960857464D1 / (0.10357D2 * t230 +
0.36231D
     #1 * cg65 + 0.88026D0 * t233 + 0.49671D0 * t235))
        cg58 = cg57
        cg66 = cg61
        t270 = sqrt(dble(t12) * t133 * t71)
        cg67 = 0.2D1 * t270
        t271 = 0.1D1 / cg66
        t272 = cg0 * t271
        t273 = 0.1D1 / cg67
        cg68 = t272 * t273 * t6 / 0.2D1
        t276 = 0.1D1 / cg58
        t278 = cg66 ** 2
        t279 = t278 * cg66
        t280 = 0.1D1 / t279
        t282 = exp(-cg41 * t276 * t280)
        t283 = t282 - 0.1D1
        cg7 = 0.66725D-1 * t276 / t283
        t286 = cg58 * t279
        t287 = cg68 ** 2
        t288 = t276 * t287
        t289 = cg7 * t287
        t290 = 0.1D1 + t289
        t291 = cg7 ** 2
        t292 = t287 ** 2
        t294 = 0.1D1 + t289 + t291 * t292
        t295 = 0.1D1 / t294
        t296 = t290 * t295
        t299 = 0.1D1 + 0.66725D-1 * t288 * t296
        t300 = log(t299)
        cg22 = cg41 + t286 * t300
        t301 = rhoa - rhob
        t302 = 0.1D1 / rho
        chi = t301 * t302
```

```
        t304 = (t71 * t302) ** (0.1D1 / 0.3D1)
        rs = dble(t141) * t304 / 0.4D1
        t307 = 0.1D1 + 0.21370D0 * rs
        t308 = sqrt(rs)
        t311 = t308 * rs
        t313 = rs ** 0.20D1
        t315 = 0.75957D1 * t308 + 0.35876D1 * rs + 0.16382D1 * t311 +
     0.
     #49294D0 * t313
        t318 = 0.1D1 + 0.1608182432D2 / t315
        t319 = log(t318)
        cg69 = -0.62182D-1 * t307 * t319
        t323 = 0.1D1 + 0.20548D0 * rs
        t328 = 0.141189D2 * t308 + 0.61977D1 * rs + 0.33662D1 * t311 +
     0
     #.62517D0 * t313
        t331 = 0.1D1 + 0.3216468318D2 / t328
        t332 = log(t331)
        t336 = 0.1D1 + 0.11125D0 * rs
        t341 = 0.10357D2 * t308 + 0.36231D1 * rs + 0.88026D0 * t311 +
     0.
     #49671D0 * t313
        t344 = 0.1D1 + 0.2960857464D1 / t341
        t345 = log(t344)
        cg70 = 0.33774D0 * t336 * t345
        t347 = 0.1D1 + chi
        t348 = t347 ** (0.1D1 / 0.3D1)
        t349 = t348 * t347
        t350 = 0.1D1 - chi
        t351 = t350 ** (0.1D1 / 0.3D1)
        t352 = t351 * t350
        f = (t349 + t352 - 0.2D1) * dble(t188)
        t354 = cg70 * f
        t355 = 0.9D1 / 0.8D1 / dble(t188)
        t356 = chi ** 2
        t357 = t356 ** 2
        t359 = t355 * (0.1D1 - t357)
        t361 = -0.31090D-1 * t323 * t332 - cg69
        t362 = t361 * f
        cg60 = cg69 + t354 * t359 + t362 * t357
        cg4 = cg58
        t364 = t348 ** 2
        t365 = t351 ** 2
        phi = t364 / 0.2D1 + t365 / 0.2D1
        t367 = t14 * rho
        t368 = t367 ** (0.1D1 / 0.3D1)
        t370 = sqrt(dble(t12) * t368 * t71)
```

```
        cg71 = 0.2D1 * t370
        t371 = 0.1D1 / phi
        t372 = my_norm_drho * t371
        t373 = 0.1D1 / cg71
        t374 = t373 * t302
        t = t372 * t374 / 0.2D1
        t376 = 0.1D1 / cg4
        t377 = cg60 * t376
        t378 = phi ** 2
        t379 = t378 * phi
        t380 = 0.1D1 / t379
        t382 = exp(-t377 * t380)
        t383 = t382 - 0.1D1
        A = 0.66725D-1 * t376 / t383
        t386 = cg4 * t379
        t387 = t ** 2
        t388 = t376 * t387
        t389 = A * t387
        t390 = 0.1D1 + t389
        t391 = A ** 2
        t392 = t387 ** 2
        t394 = 0.1D1 + t389 + t391 * t392
        t395 = 0.1D1 / t394
        t396 = t390 * t395
        t399 = 0.1D1 + 0.66725D-1 * t388 * t396
        t400 = log(t399)
        cg72 = cg60 + t386 * t400
        t403 = rhoa * rhob
        t404 = my_norm_drho ** 2
        t409 = sqrt(dble(t1) * t81 + dble(t5) * t20 - t403 * t404 +
t403
     # * dble(t1) + t403 * dble(t5))
        t410 = rho ** 2
        t411 = 0.1D1 / t410
        t413 = t12 ** 2
        t414 = 0.2D1 * t409 * t411 * dble(t413)
        t415 = 0.1D1 / t368
        eps = t414 * t415 / 0.6D1
        cg73 = 0.53D0 + 0.87D0 * t356 + 0.5D0 * t357 + 0.226D1 * t357 *
     #t356
        cg74 = t404 * t302 / 0.8D1
        t422 = eps ** 2
        t425 = 0.1D1 / t349 + 0.1D1 / t352
        t428 = 0.1D1 + t422 * t425 / 0.2D1
        t429 = t428 ** 2
        t430 = t429 ** 2
        t431 = 0.1D1 / t430
```

```
        cg42 = cg73 * t431
        ma = max(cg72, cg21)
        mb = max(cg72, cg22)
        t432 = cg72 * cg42
        t433 = cg74 ** 2
        t435 = 0.1D1 + cg42
        t436 = t435 * t433
        t437 = rhoa * t302
        t439 = rhob * t302
        t441 = t437 * ma + t439 * mb
        t443 = t432 * t433 - t436 * t441
        t444 = tau ** 2
        t445 = 0.1D1 / t444
        cg23 = cg72 + t443 * t445
        t447 = rho * cg23
        t448 = t433 * cg74
        t449 = cg23 * t448
        t451 = 0.1D1 / t444 / tau
        t454 = 0.1D1 + 0.28D1 * t449 * t451
        energy = t78 * cg31 + t137 * cg13 + t447 * t454
        cg24 = -dble(t13) * t19 / t22 / t20 / rhoa / 0.9D1
        t463 = 0.1D1 / t20
        cg43 = -dble(t1) * t463 * t27 / 0.8D1
        t469 = cg29 / t41
        cg59 = 0.5D1 / 0.3D1 * cg24 * t30 - 0.5D1 / 0.3D1 * t469 * cg43
        t476 = t32 / t36 / t35
        cg16 = 0.9D1 / 0.20D2 * cg59 * t37 - 0.9D1 / 0.40D2 * t476 *
(0.
     #4D0 * cg59 * t32 + 0.4D0 * cg35 * cg59) + 0.2D1 / 0.3D1 * cg24
        t485 = cg30 * t44
        t491 = t41 * cg30 / t43 / t42
        t502 = cg54 / t55
        t503 = cg30 * cg43
        t505 = cg29 * cg24
        t518 = t62 / t65 / t64
        t521 = t68 ** 2
        t522 = 0.1D1 / t521
        t524 = 0.3141592654D1 * dble(t12)
        t525 = t74 ** 2
        t526 = 0.1D1 / t525
        t535 = t301 * t411
        cg75 = t302 - t535
        t536 = t304 ** 2
        rsrhoa = -dble(t141) / t536 * t71 * t411 / 0.12D2
        t544 = t315 ** 2
        t546 = t307 / t544
        t547 = 0.1D1 / t308
```

```
        t548 = t547 * rsrhoa
        t551 = t308 * rsrhoa
        t553 = rs ** 0.10D1
        t554 = t553 * rsrhoa
        t557 = 0.1D1 / t318
        cg15 = -0.1328829340D-1 * rsrhoa * t319 + 0.9999999999D0 * t546
     #* (0.3797850000D1 * t548 + 0.35876D1 * rsrhoa + 0.2457300000D1 *
t
     #551 + 0.985880D0 * t554) * t557
        t563 = t328 ** 2
        t565 = t323 / t563
        t571 = 0.1D1 / t331
        t577 = t341 ** 2
        t579 = t336 / t577
        t585 = 0.1D1 / t344
        frhoa = (0.4D1 / 0.3D1 * t348 * cg75 - 0.4D1 / 0.3D1 * t351 *
cg
     #75) * dble(t188)
        t597 = t356 * chi
        t598 = t355 * t597
        t607 = t597 * cg75
        cg17 = cg15 + (0.375735750D-1 * rsrhoa * t345 - 0.9999999999D0
*
     # t579 * (0.5178500000D1 * t548 + 0.36231D1 * rsrhoa +
0.1320390000
     #D1 * t551 + 0.993420D0 * t554) * t585) * f * t359 + cg70 * frhoa
*
     # t359 - 0.4D1 * t354 * t598 * cg75 + (-0.638837320D-2 * rsrhoa *
t
     #332 + 0.1000000000D1 * t565 * (0.7059450000D1 * t548 + 0.61977D1
*
     # rsrhoa + 0.5049300000D1 * t551 + 0.1250340D1 * t554) * t571 -
cg1
     #5) * f * t357 + t361 * frhoa * t357 + 0.4D1 * t362 * t607
        t610 = 0.1D1 / t348
        t612 = 0.1D1 / t351
        cg76 = t610 * cg75 / 0.3D1 - t612 * cg75 / 0.3D1
        t615 = t368 ** 2
        cg77 = dble(t12) / t615 * t14 / 0.3D1
        t619 = 0.1D1 / t370
        t622 = my_norm_drho / t378
        t625 = cg71 ** 2
        t627 = 0.1D1 / t625 * t302
        t631 = t372 * t373 * t411
        trhoa = -t622 * t374 * cg76 / 0.2D1 - t372 * t627 * t619 * cg77
     #* t71 / 0.2D1 - t631 / 0.2D1
        t633 = t383 ** 2
```

```
        t635 = t376 / t633
        t638 = t378 ** 2
        t639 = 0.1D1 / t638
        Arhoa = -0.66725D-1 * t635 * (-cg17 * t376 * t380 + 0.3D1 *
t377
     # * t639 * cg76) * t382
        t647 = cg4 * t378
        t651 = t376 * t
        t655 = Arhoa * t387
        t656 = A * t
        t658 = 0.2D1 * t656 * trhoa
        t663 = t394 ** 2
        t665 = t390 / t663
        t666 = A * t392
        t669 = t387 * t
        t670 = t391 * t669
        t678 = 0.1D1 / t399
        cg18 = cg17 + 0.3D1 * t647 * t400 * cg76 + t386 * (0.133450D0 *
     #t651 * t396 * trhoa + 0.66725D-1 * t388 * (t655 + t658) * t395 -
0
     #.66725D-1 * t388 * t665 * (t655 + t658 + 0.2D1 * t666 * Arhoa +
0.
     #4D1 * t670 * trhoa)) * t678
        t682 = 0.1D1 / t409 * t411
        t683 = dble(t5) * rhoa
        t686 = rhob * dble(t1)
        t693 = 0.4D1 * t409 / t410 / rho
        t701 = t414 / t368 / t367 * t14 / 0.18D2
        t705 = t357 * chi
        cg44 = -t404 * t411 / 0.8D1
        t713 = cg73 / t430 / t428
        t714 = eps * t425
        t716 = t347 ** 2
        t718 = 0.1D1 / t348 / t716
        t720 = t350 ** 2
        t722 = 0.1D1 / t351 / t720
        cg78 = (0.174D1 * chi * cg75 + 0.20D1 * t607 + 0.1356D2 * t705
*
     # cg75) * t431 - 0.4D1 * t713 * (t714 * ((t682 * (0.2D1 * t683 -
rh
     #ob * t404 + t686 + rhob * dble(t5)) - t693) * dble(t413) * t415 /
     #0.6D1 - t701) + t422 * (-0.4D1 / 0.3D1 * t718 * cg75 + 0.4D1 /
0.3
     #D1 * t722 * cg75) / 0.2D1)
        t731 = cg72 .lt. cg22
        mbrhoa = myIF(t731, 0, cg18)
        t732 = t143 ** 2
```

```
        cg9 = -dble(t141) / t732 * t71 * t463 / 0.12D2
        t740 = t167 ** 2
        t749 = cg12 ** 0.10D1
        cg79 = -0.638837320D-2 * cg9 * t171 + 0.1000000000D1 * t162 /
t7
     #40 * (0.7059450000D1 / t147 * cg9 + 0.61977D1 * cg9 +
0.5049300000
     #D1 * t147 * cg9 + 0.1250340D1 * t749 * cg9) / t170
        t761 = cg62 ** 2
        cg45 = -t196 / t761 * t2 / t194 * dble(t12) * t526 * t14 * t71
/
     # 0.6D1 - t196 * t197 * t463 / 0.2D1
        t769 = cg57 ** 2
        t771 = t207 ** 2
        cg46 = 0.66725D-1 / t769 / t771 * cg79 * t204 * t206
        t777 = t200 * cg63
        t781 = cg46 * t211
        t782 = cg64 * cg63
        t784 = 0.2D1 * t782 * cg45
        t789 = t218 ** 2
        t791 = t214 / t789
        t795 = t211 * cg63
        t796 = t215 * t795
        t804 = 0.1D1 / t223
        t806 = cg72 .lt. cg21
        marhoa = myIF(t806, cg79 + t210 * (0.133450D0 * t777 * t220 *
cg
     #45 + 0.66725D-1 * t212 * (t781 + t784) * t219 - 0.66725D-1 * t212
     #* t791 * (t781 + t784 + 0.2D1 * cg64 * t216 * cg46 + 0.4D1 * t796
     #* cg45)) * t804, cg18)
        t816 = t435 * cg74
        t822 = rhoa * t411 * ma
        t825 = rhob * t411 * mb
        cg81 = cg18 + (cg18 * cg42 * t433 + cg72 * cg78 * t433 + 0.2D1
*
     # t432 * cg74 * cg44 - cg78 * t433 * t441 - 0.2D1 * t816 * t441 *
c
     #g44 - t436 * (t302 * ma - t822 + t437 * marhoa - t825 + t439 *
mbr
     #hoa)) * t445
        t831 = cg23 * t454
        t837 = cg23 * t433
        deriv_rhoa = cg36 * cg31 - rhoa * t524 * dble(t18) * t526 *
cg31
     # / 0.4D1 + 0.1000000000D1 * t78 * t522 * (((0.318192D1 * t485 *
cg
     #43 - 0.636384D1 * t491 * cg43) * cg29 + t47 * cg24 + 0.292D3 /
```

```
0.2
     #025D4 * cg54 * cg16 - 0.73D2 / 0.4050D4 * cg16 * t55 - 0.73D2 /
0.
     #8100D4 * t502 * (0.36D2 * t503 + 0.100D3 * t505) +
0.3791437570D-1
     # * t505 + 0.2204014296D0 * t503 + 0.101216061D1 * t52 * cg24) *
t6
     #6 - 0.2479516082D1 * t518 * cg24) + t831 + rho * cg81 * t454 +
t44
     #7 * (0.28D1 * cg81 * t448 * t451 + 0.84D1 * t837 * t451 * cg44)
        cg47 = -dble(t80) * t19 / t83 / t81 / rhob / 0.9D1
        t850 = 0.1D1 / t81
        cg20 = -dble(t5) * t850 * t88 / 0.8D1
        t856 = cg32 / t102
        cg80 = 0.5D1 / 0.3D1 * cg47 * t91 - 0.5D1 / 0.3D1 * t856 * cg20
        t863 = t93 / t97 / t96
        cg19 = 0.9D1 / 0.20D2 * cg80 * t98 - 0.9D1 / 0.40D2 * t863 *
(0.
     #4D0 * cg80 * t93 + 0.4D0 * cg38 * cg80) + 0.2D1 / 0.3D1 * cg47
        t872 = cg33 * t105
        t878 = t102 * cg33 / t104 / t103
        t889 = cg56 / t116
        t890 = cg33 * cg20
        t892 = cg32 * cg47
        t905 = t123 / t126 / t125
        t908 = t129 ** 2
        t909 = 0.1D1 / t908
        t911 = t133 ** 2
        t912 = 0.1D1 / t911
        cg82 = -t302 - t535
        rsrhob = rsrhoa
        t923 = t547 * rsrhob
        t926 = t308 * rsrhob
        t928 = t553 * rsrhob
        cg48 = -0.1328829340D-1 * rsrhob * t319 + 0.9999999999D0 * t546
     #* (0.3797850000D1 * t923 + 0.35876D1 * rsrhob + 0.2457300000D1 *
t
     #926 + 0.985880D0 * t928) * t557
        frhob = (0.4D1 / 0.3D1 * t348 * cg82 - 0.4D1 / 0.3D1 * t351 *
cg
     #82) * dble(t188)
        t970 = t597 * cg82
        cg84 = cg48 + (0.375735750D-1 * rsrhob * t345 - 0.9999999999D0
*
     # t579 * (0.5178500000D1 * t923 + 0.36231D1 * rsrhob +
0.1320390000
     #D1 * t926 + 0.993420D0 * t928) * t585) * f * t359 + cg70 * frhob
```

```
*
     # t359 - 0.4D1 * t354 * t598 * cg82 + (-0.638837320D-2 * rsrhob *
t
     #332 + 0.1000000000D1 * t565 * (0.7059450000D1 * t923 + 0.61977D1
*
     # rsrhob + 0.5049300000D1 * t926 + 0.1250340D1 * t928) * t571 -
cg4
     #8) * f * t357 + t361 * frhob * t357 + 0.4D1 * t362 * t970
        cg93 = t610 * cg82 / 0.3D1 - t612 * cg82 / 0.3D1
        trhob = -t622 * t374 * cg93 / 0.2D1 - t372 * t627 * t619 * cg77
     #* t71 / 0.2D1 - t631 / 0.2D1
        Arhob = -0.66725D-1 * t635 * (-cg84 * t376 * t380 + 0.3D1 *
t377
     # * t639 * cg93) * t382
        t997 = Arhob * t387
        t999 = 0.2D1 * t656 * trhob
        cg85 = cg84 + 0.3D1 * t647 * t400 * cg93 + t386 * (0.133450D0 *
     #t651 * t396 * trhob + 0.66725D-1 * t388 * (t997 + t999) * t395 -
0
     #.66725D-1 * t388 * t665 * (t997 + t999 + 0.2D1 * t666 * Arhob +
0.
     #4D1 * t670 * trhob)) * t678
        cg8 = cg44
        cg6 = (0.174D1 * chi * cg82 + 0.20D1 * t970 + 0.1356D2 * t705 *
     #cg82) * t431 - 0.4D1 * t713 * (t714 * ((t682 * (0.2D1 * t686 -
rho
     #a * t404 + rhoa * dble(t1) + t683) - t693) * dble(t413) * t415 /
0
     #.6D1 - t701) + t422 * (-0.4D1 / 0.3D1 * t718 * cg82 + 0.4D1 /
0.3D
     #1 * t722 * cg82) / 0.2D1)
        marhob = myIF(t806, 0, cg85)
        t1039 = t226 ** 2
        cg49 = -dble(t141) / t1039 * t71 * t850 / 0.12D2
        t1047 = t250 ** 2
        t1056 = cg65 ** 0.10D1
        cg55 = -0.638837320D-2 * cg49 * t254 + 0.1000000000D1 * t245 /
t
     #1047 * (0.7059450000D1 / t230 * cg49 + 0.61977D1 * cg49 +
0.504930
     #0000D1 * t230 * cg49 + 0.1250340D1 * t1056 * cg49) / t253
        t1068 = cg67 ** 2
        cg50 = -t272 / t1068 * t6 / t270 * dble(t12) * t912 * t14 * t71
     #/ 0.6D1 - t272 * t273 * t850 / 0.2D1
        t1076 = cg58 ** 2
        t1078 = t283 ** 2
        cg51 = 0.66725D-1 / t1076 / t1078 * cg55 * t280 * t282
```

```
        t1084 = t276 * cg68
        t1088 = cg51 * t287
        t1089 = cg7 * cg68
        t1091 = 0.2D1 * t1089 * cg50
        t1096 = t294 ** 2
        t1098 = t290 / t1096
        t1102 = t287 * cg68
        t1103 = t291 * t1102
        t1111 = 0.1D1 / t299
        mbrhob = myIF(t731, cg55 + t286 * (0.133450D0 * t1084 * t296 *
c
     #g50 + 0.66725D-1 * t288 * (t1088 + t1091) * t295 - 0.66725D-1 *
t2
     #88 * t1098 * (t1088 + t1091 + 0.2D1 * cg7 * t292 * cg51 + 0.4D1 *
     #t1103 * cg50)) * t1111, cg85)
        cg83 = cg85 + (cg85 * cg42 * t433 + cg72 * cg6 * t433 + 0.2D1 *
     #t432 * cg74 * cg8 - cg6 * t433 * t441 - 0.2D1 * t816 * t441 * cg8
     #- t436 * (-t822 + t437 * marhob + t302 * mb - t825 + t439 *
mbrhob
     #)) * t445
        deriv_rhob = cg39 * cg13 - rhob * t524 * dble(t18) * t912 *
cg13
     # / 0.4D1 + 0.1000000000D1 * t137 * t909 * (((0.318192D1 * t872 *
c
     #g20 - 0.636384D1 * t878 * cg20) * cg32 + t108 * cg47 + 0.292D3 /
0
     #.2025D4 * cg56 * cg19 - 0.73D2 / 0.4050D4 * cg19 * t116 - 0.73D2
/
     # 0.8100D4 * t889 * (0.36D2 * t890 + 0.100D3 * t892) +
0.3791437570
     #D-1 * t892 + 0.2204014296D0 * t890 + 0.101216061D1 * t113 * cg47)
     #* t127 - 0.2479516082D1 * t905 * cg47) + t831 + rho * cg83 * t454
     #+ t447 * (0.28D1 * cg83 * t448 * t451 + 0.84D1 * t837 * t451 *
cg8
     #)
        cg94 = cg * dble(t12) * t25 / 0.12D2
        cg95 = cg * t2 * t27 / 0.4D1
        cg86 = 0.5D1 / 0.3D1 * cg94 * t30 - 0.5D1 / 0.3D1 * t469 * cg95
        cg99 = 0.9D1 / 0.20D2 * cg86 * t37 - 0.9D1 / 0.40D2 * t476 *
(0.
     #4D0 * cg86 * t32 + 0.4D0 * cg35 * cg86) + 0.2D1 / 0.3D1 * cg94
        t1171 = cg30 * cg95
        t1173 = cg29 * cg94
        cg5 = t195 * t197 * t2 / 0.2D1
        cg87 = -0.2D1 / 0.3D1 * t713 * t714 * t682 * (0.2D1 * cg * t81
+
     # 0.2D1 * t403 * cg) * dble(t413) * t415
```

```
        cg96 = myIF(t806, t210 * (0.133450D0 * t777 * t220 * cg5 +
0.133
     #450D0 * t200 * t795 * cg64 * cg5 * t219 - 0.66725D-1 * t212 *
t791
     # * (0.2D1 * t782 * cg5 + 0.4D1 * t796 * cg5)) * t804, 0)
        cg25 = (cg72 * cg87 * t433 - cg87 * t433 * t441 - t436 * t437 *
     #cg96) * t445
        deriv_norm_drhoa = 0.1000000000D1 * t78 * t522 * (((0.318192D1
*
     # t485 * cg95 - 0.636384D1 * t491 * cg95) * cg29 + t47 * cg94 +
0.2
     #92D3 / 0.2025D4 * cg54 * cg99 - 0.73D2 / 0.4050D4 * cg99 * t55 -
0
     #.73D2 / 0.8100D4 * t502 * (0.36D2 * t1171 + 0.100D3 * t1173) +
0.3
     #791437570D-1 * t1173 + 0.2204014296D0 * t1171 + 0.101216061D1 *
t5
     #2 * cg94) * t66 - 0.2479516082D1 * t518 * cg94) + rho * cg25 *
t45
     #4 + 0.28D1 * t447 * cg25 * t448 * t451
        cg97 = cg0 * dble(t12) * t86 / 0.12D2
        cg98 = cg0 * t6 * t88 / 0.4D1
        cg88 = 0.5D1 / 0.3D1 * cg97 * t91 - 0.5D1 / 0.3D1 * t856 * cg98
        cg100 = 0.9D1 / 0.20D2 * cg88 * t98 - 0.9D1 / 0.40D2 * t863 *
(0
     #.4D0 * cg88 * t93 + 0.4D0 * cg38 * cg88) + 0.2D1 / 0.3D1 * cg97
        t1259 = cg33 * cg98
        t1261 = cg32 * cg97
        cg101 = t271 * t273 * t6 / 0.2D1
        cg89 = -0.2D1 / 0.3D1 * t713 * t714 * t682 * (0.2D1 * cg0 * t20
     #+ 0.2D1 * t403 * cg0) * dble(t413) * t415
        cg102 = myIF(t731, t286 * (0.133450D0 * t1084 * t296 * cg101 +
0
     #.133450D0 * t276 * t1102 * cg7 * cg101 * t295 - 0.66725D-1 * t288
     #* t1098 * (0.2D1 * t1089 * cg101 + 0.4D1 * t1103 * cg101)) *
t1111
     #, 0)
        cg26 = (cg72 * cg89 * t433 - cg89 * t433 * t441 - t436 * t439 *
     #cg102) * t445
        deriv_norm_drhob = 0.1000000000D1 * t137 * t909 * (((0.318192D1
     #* t872 * cg98 - 0.636384D1 * t878 * cg98) * cg32 + t108 * cg97 +
0
     #.292D3 / 0.2025D4 * cg56 * cg100 - 0.73D2 / 0.4050D4 * cg100 *
t11
     #6 - 0.73D2 / 0.8100D4 * t889 * (0.36D2 * t1259 + 0.100D3 * t1261)
     #+ 0.3791437570D-1 * t1261 + 0.2204014296D0 * t1259 +
0.101216061D1
```

```
      # * t113 * cg97) * t127 - 0.2479516082D1 * t905 * cg97) + rho *
cg2
      #6 * t454 + 0.28D1 * t447 * cg26 * t448 * t451
         cg11 = t371 * t373 * t302 / 0.2D1
         cg14 = t386 * (0.133450D0 * t651 * t396 * cg11 + 0.133450D0 *
t3
      #76 * t669 * A * cg11 * t395 - 0.66725D-1 * t388 * t665 * (0.2D1 *
      #t656 * cg11 + 0.4D1 * t670 * cg11)) * t678
         cg103 = my_norm_drho * t302 / 0.4D1
         cg90 = 0.4D1 / 0.3D1 * t713 * t714 * t682 * t403 * my_norm_drho
      #* dble(t413) * t415
         cg10 = myIF(t806, 0, cg14)
         cg28 = myIF(t731, 0, cg14)
         cg27 = cg14 + (cg14 * cg42 * t433 + cg72 * cg90 * t433 + 0.2D1
*
      # t432 * cg74 * cg103 - cg90 * t433 * t441 - 0.2D1 * t816 * t441 *
      #cg103 - t436 * (t437 * cg10 + t439 * cg28)) * t445
         deriv_norm_drho = rho * cg27 * t454 + t447 * (0.28D1 * cg27 *
t4
      #48 * t451 + 0.84D1 * t837 * t451 * cg103)
         t1376 = my_tau_a ** 2
         cg52 = -cg34 / t1376 / 0.2D1
         cg104 = -0.5D1 / 0.3D1 * t469 * cg52

         cg91 = 0.9D1 / 0.20D2 * cg104 * t37 - 0.9D1 / 0.40D2 * t476 *
(0
      #.4D0 * cg104 * t32 + 0.4D0 * cg35 * cg104)
         t1401 = cg30 * cg52
         cg105 = -0.2D1 * t443 * t451
         t1415 = t444 ** 2
         t1418 = 0.84D1 * t449 / t1415
         deriv_tau_a = 0.1000000000D1 * t78 * t522 * ((0.318192D1 * t485
      #* cg52 - 0.636384D1 * t491 * cg52) * cg29 + 0.292D3 / 0.2025D4 *
c
      #g54 * cg91 - 0.73D2 / 0.4050D4 * cg91 * t55 - 0.73D2 / 0.225D3 *
t
      #502 * t1401 + 0.2204014296D0 * t1401) * t66 + rho * cg105 * t454
+
      # t447 * (0.28D1 * cg105 * t448 * t451 - t1418)
         t1422 = my_tau_b ** 2
         cg53 = -cg37 / t1422 / 0.2D1
         cg107 = -0.5D1 / 0.3D1 * t856 * cg53
         cg92 = 0.9D1 / 0.20D2 * cg107 * t98 - 0.9D1 / 0.40D2 * t863 *
(0
      #.4D0 * cg107 * t93 + 0.4D0 * cg38 * cg107)
         t1447 = cg33 * cg53
         cg106 = cg105
```

```
        deriv_tau_b = 0.1000000000D1 * t137 * t909 * ((0.318192D1 *
t872
     # * cg53 - 0.636384D1 * t878 * cg53) * cg32 + 0.292D3 / 0.2025D4 *
     #cg56 * cg92 - 0.73D2 / 0.4050D4 * cg92 * t116 - 0.73D2 / 0.225D3
*
     # t889 * t1447 + 0.2204014296D0 * t1447) * t127 + rho * cg106 *
t45
     #4 + t447 * (0.28D1 * cg106 * t448 * t451 - t1418)
        cg108 = deriv_tau_b
        return
      end

>
```