

The present worksheet contains the formulae necessary to calculate the PBE functional, and to generate code that calculate it efficiently in different languages (fortran for example).

They are distributed under GPL v2 (<http://www.gnu.org/copyleft/gpl.html>) as part of the CP2K Project.

Fawzi Mohamed

PBE

Reference: John P. Perdew, Kieron Burke, Matthias Ernzerhof, Phys. Rev. Letter, vol. 77, n 18, pp. 3865-3868, 28. oct. (1996)

```
> 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):

> loc2:=eqs -> indets(map(lhs,eqs),symbol):

> 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<100000) do
  dep:=false;
  j:=i+1;
  while (j<=nops(eqs) and ii<100000) 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:

```

```

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

```

```

> 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)),eq);

eq3:=[seq(subs(result=convert(cat(lhs(eqs[nops(eqs)]),"_",arg_names[i]),
symbol),eq2[i]),i=1..nops(arg_names))];
end proc:

> with(CodeGeneration);
[C, Fortran, IntermediateCode, Java, LanguageDefinition, Matlab, Names, Save, Translate, VisualBasic]

```

exchange

[exchange energy (LDA)

```

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

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


```

[Uniform gas exchange:

```

> eqx2:=kf=(3*Pi^2*rho)^(1/3);
eqx3:=ex_unif=-3/(4*Pi)*kf;

$$eqx2 := kf = 3^{(1/3)} (\pi^2 \rho)^{(1/3)}$$


$$eqx3 := ex\_unif = - \frac{3 \, kf}{4 \, \pi}$$


```

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

```

> eqx4:=p=norm_drho^2/(4*(3*Pi^2)^(2/3)*rho^(8/3));
eqx5:=s=norm_drho/(2*kf*rho);

```

$$eqx4 := p = \frac{norm_drho^2 3^{(1/3)}}{12 (\pi^2)^{(2/3)} \rho^{(8/3)}}$$

$$eqx5 := s = \frac{norm_drho}{2 kf \rho}$$

```
> evalb(simplify(subs(eqx4,eqx5,eqx2,s^2=p),symbolic));
true
```

Fx can be written as

```
> eqx6:=Fx=1+kappa-kappa/(1+mu*s^2/kappa);
eqk1:=kappa=0.804;
eqpbe8:=beta=0.066725;
eqk2:=mu=beta*(Pi^2/3);
```

$$eqx6 := Fx = 1 + \kappa - \frac{\kappa}{1 + \frac{\mu s^2}{\kappa}}$$

$$eqk1 := \kappa = 0.804$$

$$eqpbe8 := \beta = 0.066725$$

$$eqk2 := \mu = \frac{1}{3} \beta \pi^2$$

```
> eqs_ex_lda := [eqk1,eqpbe8,eqk2,eqx2,eqx3,eqx5, eqx6, eqx1];
```

$$eqs_ex_lda := \left[\kappa = 0.804, \beta = 0.066725, \mu = \frac{1}{3} \beta \pi^2, kf = 3^{(1/3)} (\pi^2 \rho)^{(1/3)}, ex_unif = -\frac{3 kf}{4 \pi}, \right. \\ \left. s = \frac{norm_drho}{2 kf \rho}, Fx = 1 + \kappa - \frac{\kappa}{1 + \frac{\mu s^2}{\kappa}}, ex_lda = \rho ex_unif Fx \right]$$

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

$$\{\pi, norm_drho, \rho\}$$

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

$$\{s, ex_lda, ex_unif, Fx, kf, \mu, \kappa, \beta\}$$

correlation

```
> eqc1:=ec=rho*epsilon_cGGA;
```

$$eqc1 := ec = \rho \epsilon_{cGGA}$$

```
> eqc3:=chi=(rhoa-rhob)/rho;
```

$$eqc3 := \chi = \frac{\rho_{ho} - \rho_{hb}}{\rho}$$

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

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

PBE (alias epsilon_cGGA) from Perdew, Burke, Ernzerhof, PRL, vol 77, p 3865 (1996) It has some corrections and discussions.

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

$$eqpbe1 := t = \frac{\text{norm_drho}}{2 \phi k_s \rho}$$

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

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

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

$$eqpbe3 := k_s = 2 \sqrt{\frac{k_f}{\pi}}$$

> 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));

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

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

$$eqpbe6 := A = \frac{\beta}{\text{gamma_var} \left(e^{\left(-\frac{\text{epsilon_c_unif}}{\text{gamma_var} \phi^3} \right)} - 1 \right)}$$

> eqpbe7:=epsilon_cGGA=epsilon_c_unif+H;

$$eqpbe7 := \text{epsilon_cGGA} = \text{epsilon_c_unif} + H$$

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

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

0.03109069086

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

$$eqpbe10 := k_f = 3^{(1/3)} (\pi^2 \rho)^{(1/3)}$$

```
> eqs_pbec1 := [eqpbe8,eqpbe9,eqc3, eqpbe2, eqpbe10, eqpbe3, eqpbe1,
eqpbe6, eqpbe5,eqpbe7,eqc1];
```

$$\text{eqs_pbec1} := \left[\begin{aligned} &\beta = 0.066725, \gamma_{\text{var}} = \frac{1 - \ln(2)}{\pi^2}, \chi = \frac{\rho_{\text{hoa}} - \rho_{\text{hob}}}{\rho}, \\ &\varphi = \frac{1}{2} (1 + \chi)^{(2/3)} + \frac{1}{2} (1 - \chi)^{(2/3)}, k_f = 3^{(1/3)} (\pi^2 \rho)^{(1/3)}, k_s = 2 \sqrt{\frac{k_f}{\pi}}, t = \frac{\text{norm_drho}}{2 \varphi k_s \rho}, \\ &A = \frac{\beta}{\gamma_{\text{var}} \left(e^{\left(-\frac{\epsilon_{\text{c_unif}}}{\gamma_{\text{var}} \varphi^3} \right)} - 1 \right)}, \\ &H = \gamma_{\text{var}} \varphi^3 \ln \left(1 + \frac{\beta t^2 (1 + A t^2)}{\gamma_{\text{var}} (1 + A t^2 + A^2 t^4)} \right), \\ &\epsilon_{\text{cGGA}} = \epsilon_{\text{c_unif}} + H, \epsilon_{\text{c}} = \rho \epsilon_{\text{cGGA}} \end{aligned} \right]$$

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

$\{\pi, \text{norm_drho}, \rho, \rho_{\text{hoa}}, \rho_{\text{hob}}, \epsilon_{\text{c_unif}}\}$

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;
```

$$\text{equc1} := \epsilon_{\text{c_unif}} = e_{\text{c_u_0}} + \frac{\alpha_{\text{c}} f (1 - \chi^4)}{f_{\text{ii_0}}} + (e_{\text{c_u_1}} - e_{\text{c_u_0}}) f \chi^4$$

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

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

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

$$\text{equc3} := f_{\text{ii_0}} = \frac{8}{9 (2 \cdot 2^{(1/3)} - 2)}$$

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))));
```

$$G_{uc} := -2 A (1 + \alpha_1 rs) \ln \left(1 + \frac{1}{2 A (\beta_1 \sqrt{rs} + \beta_2 rs + \beta_3 rs^{(3/2)} + \beta_4 rs^{(p+1)})} \right)$$

> **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 := {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}

$$equc7 := e_{c_u_1} = -0.031090 (1 + 0.20548 rs) \ln \left(1 + \frac{32.16468318}{14.1189 \sqrt{rs} + 6.1977 rs + 3.3662 rs^{(3/2)} + 0.62517 rs^{2.0}} \right)$$

> **equc8:={p=1.0,A=0.16887,alpha_1=0.11125,beta_1=10.357,beta_2=3.6231,beta_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];**

$$eqs_e_c_unif := \left[\chi = \frac{\rho_{hoa} - \rho_{hob}}{\rho}, rs = \frac{1}{4} 3^{(1/3)} 4^{(2/3)} \left(\frac{1}{\pi \rho} \right)^{(1/3)}, e_{c_u_0} = -0.062182 (1 + 0.21370 rs) \ln \left(1 + \frac{16.08182432}{7.5957 \sqrt{rs} + 3.5876 rs + 1.6382 rs^{(3/2)} + 0.49294 rs^{2.0}} \right), e_{c_u_1} = \right]$$

$$0.031090 (1 + 0.20548 rs) \ln \left(1 + \frac{32.16468318}{14.1189 \sqrt{rs} + 6.1977 rs + 3.3662 rs^{(3/2)} + 0.62517 rs^{2.0}} \right)$$

$$alpha_c = 0.33774 (1 + 0.11125 rs) \ln \left(1 + \frac{2.960857464}{10.357 \sqrt{rs} + 3.6231 rs + 0.88026 rs^{(3/2)} + 0.49671 rs^{2.0}} \right), f_{\ddot{u}0} = \frac{8}{9 (2^{(1/3)} - 2)},$$

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

$$epsilon_c_unif = e_c_u_0 + \frac{alpha_c f (1 - \chi^4)}{f_{\ddot{u}0}} + (e_c_u_1 - e_c_u_0) f \chi^4 \Bigg]$$

> unk(eqs_e_c_unif);

{ $\pi, \rho, \rho_{hoa}, \rho_{hob}$ }

> loc(eqs_e_c_unif)intersect loc(eqs_pbec1);

{ χ }

> 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)];

$$eqs_pbec2 := \left[\chi = \frac{\rho_{hoa} - \rho_{hob}}{\rho}, rs = \frac{1}{4} 3^{(1/3)} 4^{(2/3)} \left(\frac{1}{\pi \rho} \right)^{(1/3)}, e_c_u_0 = -0.062182 (1 + 0.21370$$

$$\ln \left(1 + \frac{16.08182432}{7.5957 \sqrt{rs} + 3.5876 rs + 1.6382 rs^{(3/2)} + 0.49294 rs^{2.0}} \right), e_c_u_1 = -0.031090 (1 + 0.20548 rs) \ln \left(1 + \frac{32.16468318}{14.1189 \sqrt{rs} + 6.1977 rs + 3.3662 rs^{(3/2)} + 0.62517 rs^{2.0}} \right), alpha_c :$$

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

$$f_{\ddot{u}0} = \frac{8}{9 (2^{(1/3)} - 2)}, f = \frac{(1 + \chi)^{(4/3)} + (1 - \chi)^{(4/3)} - 2}{2^{(1/3)} - 2},$$

$$epsilon_c_unif = e_c_u_0 + \frac{alpha_c f (1 - \chi^4)}{f_{\ddot{u}0}} + (e_c_u_1 - e_c_u_0) f \chi^4, \beta = 0.066725,$$

$$gamma_var = \frac{1 - \ln(2)}{\pi^2}, \varphi = \frac{1}{2} (1 + \chi)^{(2/3)} + \frac{1}{2} (1 - \chi)^{(2/3)}, k_f = 3^{(1/3)} (\pi^2 \rho)^{(1/3)}, k_s = 2 \sqrt{\frac{k_{\omega}}{\pi}}$$

$$t = \frac{\text{norm_drho}}{2 \varphi k_s \rho}, A = \frac{\beta}{\text{gamma_var} \left(e^{\left(-\frac{\text{epsilon_c_unif}}{\text{gamma_var} \varphi^3} \right)} - 1 \right)},$$

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

$$\text{epsilon_cGGA} = \text{epsilon_c_unif} + H, \text{ec} = \rho \text{epsilon_cGGA}$$

> unk(eqs_pbec2);

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

> loc(eqs_pbec2);

$$\{f, t, A, \text{epsilon_cGGA}, \text{ec}, \chi, rs, \varphi, k_s, k_f, \beta, \text{gamma_var}, H, \text{epsilon_c_unif}, e_c_u_0, \alpha_c, f_ii_0, e_c_u_1\}$$

— Tests

> loc(eqs_pbec2) intersect loc(eqs_ex_lda);

$$\{\beta\}$$

> sameNameSameDef(eqs_pbec2, eqs_ex_lda);

true

> eqs_fxc := combineEqs([eqs_pbec2, eqs_ex_lda, [Fxc = Fx + epsilon_cGGA/ex_unif]]);
unk(eqs_fxc);

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

> eqs_fxc1 := subsop(indiceDef(s, eqs_fxc) = NULL, indiceDef(chi, eqs_fxc) = NULL, indiceDef(rs, eqs_fxc) = NULL, eqs_fxc);
unk(eqs_fxc1);

$$\{\pi, s, \rho, \text{norm_drho}, \chi, rs\}$$

> v_rho := solve(getDef(rs, eqs_fxc), rho);

$$v_rho := \frac{3}{4 rs^3 \pi}$$

> getDef(chi, eqs_fxc);

$$\chi = \frac{\text{rhoa} - \text{rhob}}{\rho}$$

> v_ndrho := solve(subs(getDef(kf, eqs_fxc), getDef(s, eqs_fxc)), norm_drho);

$$v_ndrho := 2 s (\pi^2 \rho)^{(1/3)} \rho 3^{(1/3)}$$

```

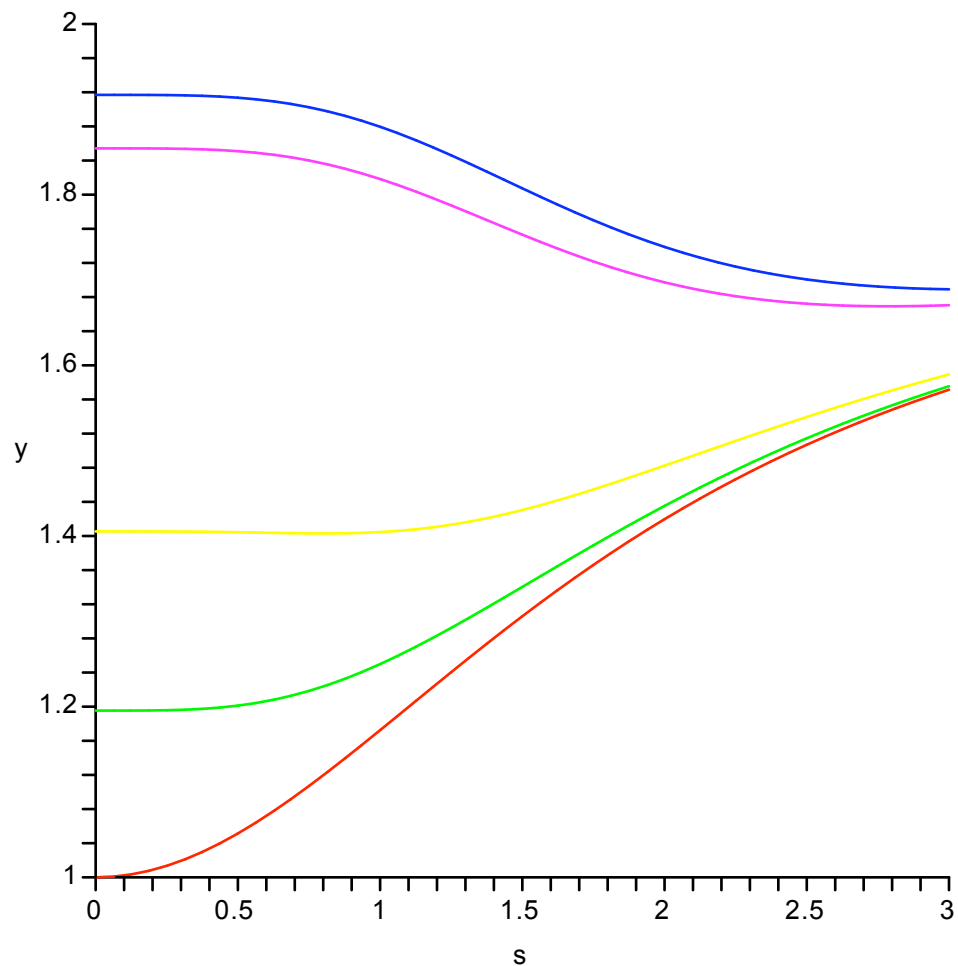
> eqs_fxc2:=[rho=v_rho,norm_drho=v_ndrho,chi=0,op(eqs_fxc1)]:
unk(eqs_fxc2);

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

> cs_eqs_fxc:=CompSeq(locals=loc(eqs_fxc2),
  globals=[Pi],params=[rs,s],eqs_fxc2):
r_eqs_fxc:=convert(cs_eqs_fxc,procedure):

> plot([r_eqs_fxc(0.,s),r_eqs_fxc(2.,s),r_eqs_fxc(10.,s),r_eqs_fxc(1.0e
4,s),r_eqs_fxc(1.0e3,s)],s=0..3,y=1..2);

```



```

> eqs_fxc_p:=combineEqs([eqs_pbec2,subs(rho=2*rho,norm_drho=2*norm_drho
,eqs_ex_lda),
  [Fxc=Fx*2^(1/3)+2^(1/3)*epsilon_cGGA/ex_unif]]):
unk(eqs_fxc_p);

                                     { $\pi$ ,  $\rho$ ,  $\rho_{ho}$ ,  $\rho_{ho}b$ ,  $norm\_drho$ }

> eqs_fxc1_p:=subsop(indiceDef(s,eqs_fxc_p)=NULL,indiceDef(chi,eqs_fxc_
p)=NULL,indiceDef(rs,eqs_fxc_p)=NULL,eqs_fxc_p):
unk(eqs_fxc1_p);

```

$$\{\pi, s, \rho, \text{norm_drho}, \chi, rs\}$$

```
> v_rho_p:=solve(getDef(rs,eqs_fxc_p),rho);
```

$$v_rho_p := \frac{3}{4 rs^3 \pi}$$

```
> getDef(chi,eqs_fxc_p);
```

$$\chi = \frac{\rho_{ho} - \rho_{hb}}{\rho}$$

```
> v_ndrho_p:=solve(subs(getDef(kf,eqs_fxc_p),getDef(s,eqs_fxc_p)),norm_
drho);
```

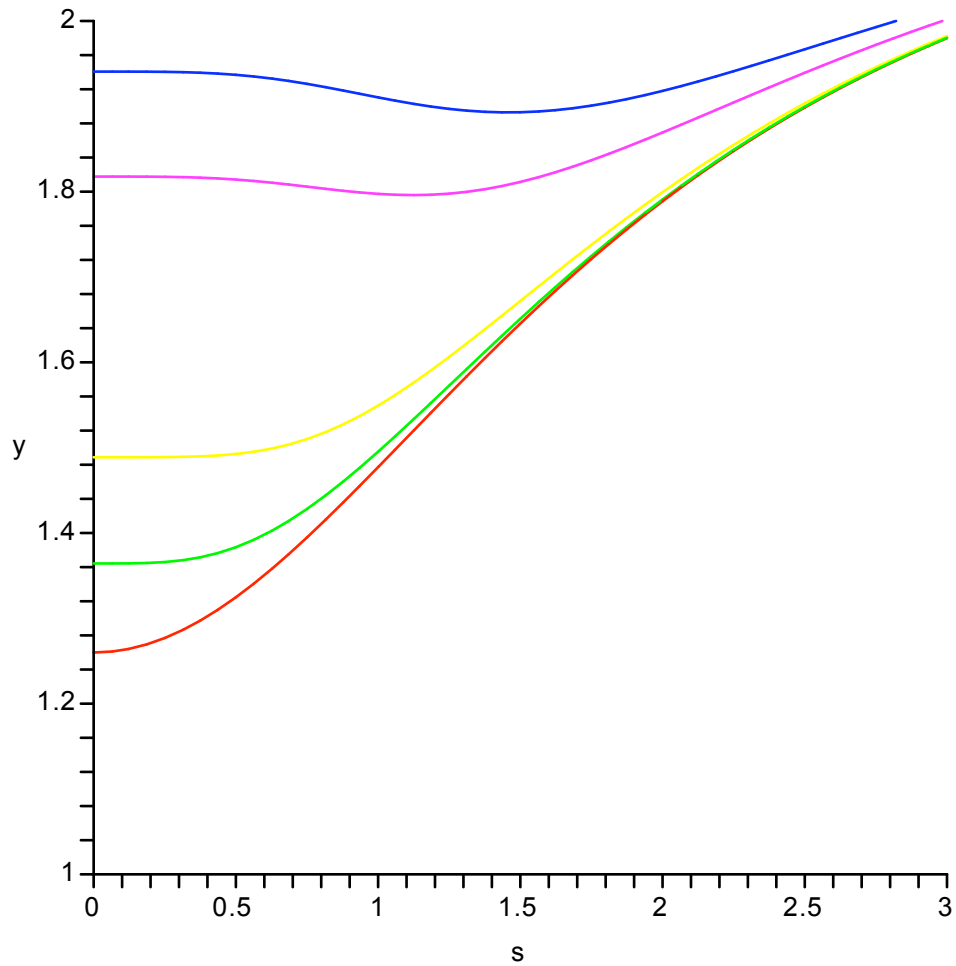
$$v_ndrho_p := 2 s (\pi^2 \rho)^{(1/3)} \rho 3^{(1/3)} 2^{(1/3)}$$

```
> eqs_fxc2_p:=[rho=v_rho_p,norm_drho=v_ndrho_p,chi=1,op(eqs_fxc1_p)]:
unk(eqs_fxc2_p);
```

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

```
> cs_eqs_fxc_p:=CompSeq(locals=loc(eqs_fxc2_p),
globals=[Pi],params=[rs,s],eqs_fxc2_p):
r_eqs_fxc_p:=convert(cs_eqs_fxc_p,procedure):
```

```
> plot([r_eqs_fxc_p(0.,s),r_eqs_fxc_p(2.,s),r_eqs_fxc_p(10.,s),r_eqs_fx
c_p(10000.,s),r_eqs_fxc_p(200000.,s)],
s=0..3,y=1..2);
```



— Compare with old QS PBE

```
> eqsPbex:=[kappa = 0.804,cx_vwn_e=-3/4*(3/Pi)^(1/3)
,f13=1/3
,r2kf = 1/2*(3*Pi^2)^(-f13)
,mu = 0.2195149727645171
,rho13 = rho^f13
,rho43 = rho13*rho
,s = r2kf*drho/rho43
,p = 1/(1 + mu*s*s/kappa)
,fx = 1 + kappa*(1 - p)
,dfx = 2*mu*s*p*p
,ex = cx_vwn_e*rho43*fx
,vx = cx_vwn_v*rho13*(fx - s*dfx)
,vxg = cx_vwn_e*r2kf*dfx/drho,energy=ex];
```

$$eqsPbex := \left[\kappa = 0.804, cx_vwn_e = -\frac{3}{4} 3^{(1/3)} \left(\frac{1}{\pi} \right)^{(1/3)}, f13 = \frac{1}{3}, r2kf = \frac{1}{2} (3 \pi^2)^{(-f13)}, \right.$$

$$\mu = 0.2195149727645171, \rho l3 = \rho^{f l3}, \rho43 = \rho l3 \rho, s = \frac{r2kf drho}{\rho43}, p = \frac{1}{1 + \frac{\mu s^2}{\kappa}},$$

$$fx = 1 + \kappa (1 - p), dfx = 2 \mu s p^2, ex = cx_vwn_e \rho43 fx, vx = cx_vwn_v \rho l3 (fx - s dfx)$$

$$vxg = \frac{cx_vwn_e r2kf dfx}{drho}, energy = ex$$

> unk(eqsPbex);

{ π , drho, ρ }

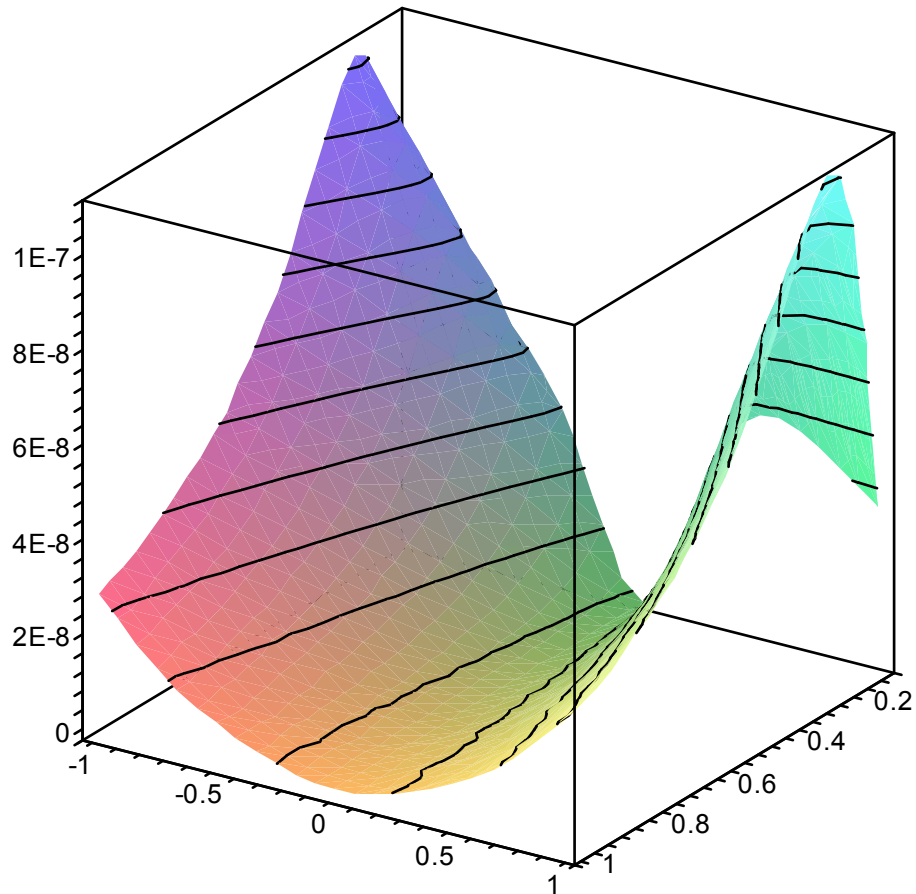
> cs_eqsPbex:=CompSeq(locals=convert(loc(eqsPbex),list),
globals=[Pi],params=[rho,drho],eqsPbex):
r_eqsPbex:=convert(cs_eqsPbex,procedure):

> eqs_ex_lda;

$$\kappa = 0.804, \beta = 0.066725, \mu = \frac{1}{3} \beta \pi^2, kf = 3^{(1/3)} (\pi^2 \rho)^{(1/3)}, ex_unif = -\frac{3 kf}{4 \pi}, s = \frac{norm_drl}{2 kf \rho}$$

$$Fx = 1 + \kappa - \frac{\kappa}{1 + \frac{\mu s^2}{\kappa}}, ex_lda = \rho ex_unif Fx$$

> cs_eqsPbex2:=CompSeq(locals=convert(loc(eqs_ex_lda),list),
globals=[Pi],params=[rho,norm_drho],eqs_ex_lda):
r_eqsPbex2:=convert(cs_eqsPbex2,procedure):
> contourplot3d(r_eqsPbex-r_eqsPbex2,0.1..1,-1..1);



```
> mu=sost([op(eqs_ex_lda),my_m=mu]),mu2=sost([op(eqsPbex),my_m=mu]);
       $\mu = 0.02224166667 \pi^2, \mu_2 = 0.2195149727645171$ 
> evalf(rhs(%[1])-rhs(%[2])),evalf((rhs(%[1])-rhs(%[2]))/rhs(%[1]));
      0.0000014785, 0.000006735258299
```

This uses VWN correlation instead of the (correct) PW92

```
> eqsPbec:=[f43=4/3
,f13=1/3
,f76=7/6
,r2ks=1/4*(3/Pi)^(-1/6)
,ap = 0.0621814
,bp = 3.72744
,cp = 12.9352
,xp = -0.10498
,beta_pbe=0.66725e-1
,gamma_pbe=(1-log(2))/Pi^2
,rsfac = (f43*Pi)^(-f13)
,x2 = rsfac*rho^(-f13)
```

```

,x = sqrt(x2)
, xp2 = xp*xp
, xmxp = x - xp
, xmxp2 = xmxp*xmxp
, pp = x*x + bp*x + cp
, qp = sqrt(4*cp - bp*bp)
, ecp = 1/2*ap*(log(x2/xmxp2) - ((xp2 + cp)*log(pp/xmxp2) +
2*bp*(xp2 - cp)*arctan(qp/(2*x + bp))/qp)/(xp2 + bp*xp + cp))
, decp = 1/2*f13*ap*((1 + bp/xmxp)*x2/pp - 1)
, ec = ecp*rho
, vc = ecp + decp
, e_var = ec/rho
, rhom76 = rho^(-f76)
, t = r2ks*drho*rhom76
, t2 = t*t
, expe = exp(-e_var/gamma_pbe)
, bog = beta_pbe/gamma_pbe
, a = bog/(expe - 1)
, at2 = a*t2
, q = 1/(1 + at2 + at2*at2)
, q2 = q*q
, p = 1 + bog*t2*(1 + at2)*q
, h = gamma_pbe*log(p)
, dpdt = 2*bog*t*(1 + 2*at2)*q2
, dtldr = -f76*t/rho
, dpda = -bog*at2*t2*t2*(2 + at2)*q2
, dade = a*a*expe/beta_pbe
, dedr = (vc - e_var)/rho
, dpdr = dpdt*dtldr + dpda*dade*dedr
, dhdr = gamma_pbe*dpdr/p
, dtddr = t/drho^2
, dhddr = gamma_pbe*dpdt*dtddr/p
, ec = ec + rho*h
, vc = vc + h + rho*dhdr
, vcg = rho*dhddr,energy=ec];

```

$$eqsPbec := \left[\begin{aligned} f43 &= \frac{4}{3}, f13 = \frac{1}{3}, f76 = \frac{7}{6}, r2ks = \frac{3^{(5/6)}}{12 \left(\frac{1}{\pi} \right)^{(1/6)}, ap = 0.0621814, bp = 3.72744, \\ cp &= 12.9352, xp = -0.10498, beta_pbe = 0.066725, gamma_pbe = \frac{1 - \ln(2)}{\pi^2}, \\ rsfac &= (f43 \pi)^{(f13)}, x2 = rsfac \rho^{(f13)}, x = \sqrt{x2}, xp2 = xp^2, xmxp = x - xp, xmxp2 = xmxp \\ pp &= x^2 + bp x + cp, qp = \sqrt{4 cp - bp^2}, \end{aligned} \right.$$

$$dec p = \frac{1}{2} fl 3 \text{ ap } \left(\frac{\left(1 + \frac{bp}{xm xp} \right) x2}{pp} - 1 \right), ec = ec p \rho, vc = ec p + dec p, e_var = \frac{ec}{\rho},$$

$$\begin{aligned} rhom76 &= \rho^{(-f76)}, t = r2ks \ drho \ rhom76, t2 = t^2, expe = \mathbf{e}^{\left(-\frac{e_var}{gamma_pbe}\right)}, \\ bog &= \frac{beta_pbe}{gamma_pbe}, a = \frac{bog}{expe - 1}, at2 = a \ t2, q = \frac{1}{1 + at2 + at2^2}, q2 = q^2, \end{aligned}$$

$$p = 1 + \log_2(1 + at^2)q, h = \gamma_{pbe} \ln(p), dp/dt = 2 \log_2 t(1 + 2at^2)q^2, dtdr = -\frac{J}{2} \frac{dp}{p} + \frac{J}{2} \frac{dt}{t} + \frac{J}{2} \frac{dr}{r}$$

$$dpda = -\log at2 \, t2^2 (2 + at2) \, q2, \, dade = \frac{a^2 \, expe}{beta \, pbe}, \, dedr = \frac{vc - e_var}{\rho},$$

$$dpdr = dpdt \, dt dr + dpda \, da de \, dedr, \, dhdr = \frac{\gamma_{pbe} \, dpdr}{p}, \, dt ddr = \frac{t}{drho^2},$$

$$dhddr = \frac{\text{gamma_pbe } dpdt \text{ } dtddr}{p}, \text{ec} = \text{ec} + \rho \text{ } h, \text{vc} = \text{vc} + h + \rho \text{ } dhdr, \text{vcg} = \rho \text{ } dhddr,$$

$$energy = ec$$

```
> cs_eqsPbec:=CompSeq(locals=convert(loc(eqsPbec),list),
    globals=[Pi],params=[rho,drho],eqsPbec):
r_eqsPbec:=convert(cs_eqsPbec,procedure):
```

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

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

```
> eqs_pbec2_lda:=[rhoa=rho/2,rhob=rho/2,op(eqs_pbec2)];
```

$$eqs_pbec2_lda := \left[\begin{array}{l} rhoa = \frac{1}{2} \rho, rhob = \frac{1}{2} \rho, \chi = \frac{rhoa - rhob}{\rho}, rs = \frac{1}{4} 3^{(1/3)} 4^{(2/3)} \left(\frac{1}{\pi \rho} \right)^{(1/3)}, \end{array} \right.$$

$$u_{-0} = -0.062182 (1 + 0.21370 r_s) \ln \left(1 + \frac{r_s}{2} \right)$$

$$\begin{aligned}
& + \frac{16.08182432}{7.5957 \sqrt{rs} + 3.5876 rs + 1.6382 rs^{(3/2)} + 0.49294 rs^{2.0}} \Bigg), e_{c_u_I} = -0.031090 (1 \\
& + 0.20548 rs) \ln \left(1 + \frac{32.16468318}{14.1189 \sqrt{rs} + 6.1977 rs + 3.3662 rs^{(3/2)} + 0.62517 rs^{2.0}} \right), alp \\
& 0.33774 (1 + 0.11125 rs) \ln \left(1 + \frac{2.960857464}{10.357 \sqrt{rs} + 3.6231 rs + 0.88026 rs^{(3/2)} + 0.49671 rs^{2.0}} \right), \\
& f_{ii_0} = \frac{8}{9 (2^{(1/3)} - 2)}, f = \frac{(1 + \chi)^{(4/3)} + (1 - \chi)^{(4/3)} - 2}{2^{(1/3)} - 2}, \\
& epsilon_c_unif = e_{c_u_0} + \frac{alpha_c f (1 - \chi^4)}{f_{ii_0}} + (e_{c_u_I} - e_{c_u_0}) f \chi^4, \beta = 0.066725, \\
& gamma_var = \frac{1 - \ln(2)}{\pi^2}, \varphi = \frac{1}{2} (1 + \chi)^{(2/3)} + \frac{1}{2} (1 - \chi)^{(2/3)}, k_f = 3^{(1/3)} (\pi^2 \rho)^{(1/3)}, \\
& k_s = 2 \sqrt{\frac{k_f}{\pi}}, t = \frac{norm_drho}{2 \varphi k_s \rho}, A = \frac{\beta}{gamma_var \left(e^{\left(\frac{- epsilon_c_unif}{gamma_var \varphi^3} \right)} - 1 \right)}, \\
& H = gamma_var \varphi^3 \ln \left(1 + \frac{\beta t^2 (1 + A t^2)}{gamma_var (1 + A t^2 + A^2 t^4)} \right), \\
& epsilon_cGGA = epsilon_c_unif + H, ec = \rho epsilon_cGGA
\end{aligned}$$

```

> cs_eqsPbec2:=CompSeq(locals=convert(loc(eqs_pbec2_lda),list),
  gobals=[Pi],params=[rho,norm_drho],eqs_pbec2_lda):
  r_eqsPbec2:=convert(cs_eqsPbec2,procedure):

```

```

> with(plots);

```

Warning, the name changecoords has been redefined

[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, replot, rootlocus, semilogplot, setoptions, setoption: spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d, tubeplot]

```

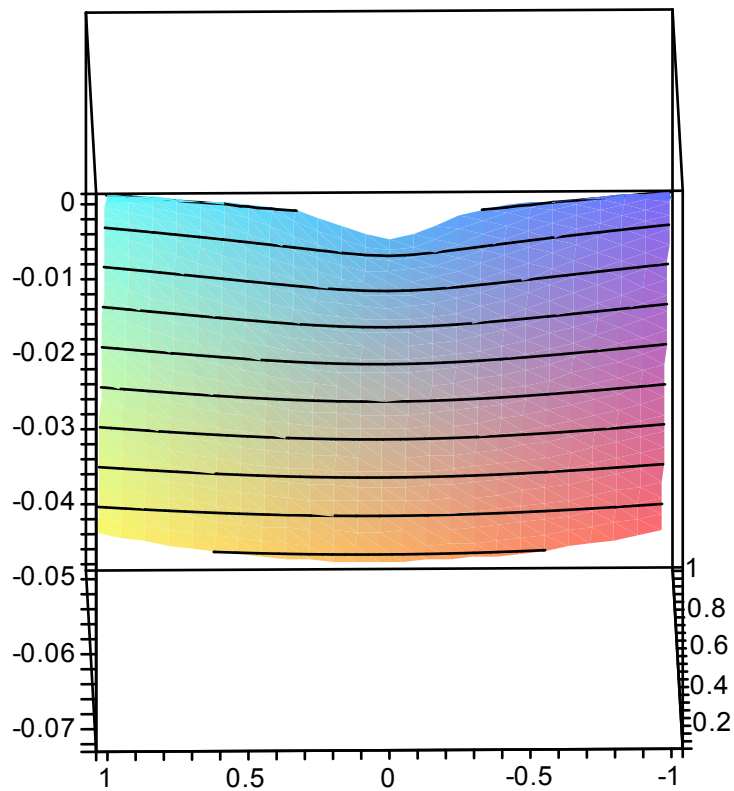
> convert(proc(r) local t; global Pi; t:=Pi*r^2; end proc,CompSeq);
  CompSeq(locals = [t], globals = [Pi], params = [r], [t = pi r^2])
> convert(CompSeq(locals=[t],params=[r],[t=r^2]),procedure);

```

```
proc(r) local t; t := r^2 end proc;
```

```
> contourplot3d(r_eqsPbec-r_eqsPbec2,0.1..1,-1..1);
```

```
> contourplot3d(r_eqsPbec,0.1..1,-1..1);
```



>

LDA

```
> loc(eqs_ex_lda)intersect loc(eqs_pbec2);
                                   { $\beta$ }

> sameNameSameDef(eqs_pbec2,eqs_ex_lda);
                                   true

> eqs_lda:=combineEqs([subs(rhoa=rho/2,rhob=rho/2,eqs_pbec2),eqs_ex_lda,[
exc=ex_lda+ec]]):
unk(eqs_lda);
                                   { $\pi$ , norm_drho,  $\rho$ }

> arg_lda_names:=[rho,norm_drho];
                                   arg_lda_names := [ $\rho$ , norm_drho]

> deriv_lda1:=calcDerivs(eqs_lda,arg_lda_names):
```

```

deriv_lda2:=[seq(op(calcDerivs(deriv_lda1[i],[rho])),i=1..2),op(calcDerivs(deriv_lda1[2],[norm_drho]))]:
deriv_lda3:=[seq(op(calcDerivs(deriv_lda2[i],[rho])),i=1..3),op(calcDerivs(deriv_lda2[3],[norm_drho]))]:

```

```

> eqs_lda2:=sostConst(eqs_lda):
sameNameSameDef(eqs_lda2,deriv_lda1[i])$i=1..2;
sameNameSameDef(eqs_lda2,deriv_lda2[i])$i=1..3;
sameNameSameDef(eqs_lda2,deriv_lda3[i])$i=1..4;
seq(sameNameSameDef(deriv_lda1[i],deriv_lda2[j])$i=1..2,j=1..3);
seq(sameNameSameDef(deriv_lda1[i],deriv_lda3[j])$i=1..2,j=1..4);
seq(sameNameSameDef(deriv_lda2[i],deriv_lda3[j])$i=1..3,j=1..4);

```

true, true

true, true, true

true, true, true, true

true, true, true, true, true, true

true, true, true, true, true, true, true

true, true, true, true, true, true, true, true, true, true, true

```

> eqs_lda3:=combineEqs([eqs_lda2,op(deriv_lda1),op(deriv_lda2),op(deriv_lda3)]):

```

```

> eqs_lda4:=enforceDependencies([my_rho=rho,my_norm_drho=norm_drho,
op(subs(rho=my_rho,norm_drho=my_norm_drho,eqs_lda3))]):

```

```

> res_eqs_lda:={exc,exc_rho,exc_norm_drho,exc_rho_rho,exc_norm_drho_rho,exc_norm_drho_norm_drho,

```

```

exc_rho_rho_rho,exc_norm_drho_rho_rho,exc_norm_drho_norm_drho_rho,
exc_norm_drho_norm_drho_norm_drho};

```

```

for my_symb in res_eqs_lda do

```

```

    print(my_symb,unk([op(eqs_lda4),result=my_symb]));

```

```

end do;

```

```

res_eqs_lda:={exc_norm_drho,exc_norm_drho_rho_rho,exc_norm_drho_norm_drho_rho,exc_rho,
exc,exc_norm_drho_norm_drho,exc_rho_rho_rho,exc_norm_drho_rho,exc_rho_rho,
exc_norm_drho_norm_drho_norm_drho}

```

exc_norm_drho, { π , norm_drho, ρ }

exc_norm_drho_rho_rho, { π , norm_drho, ρ }

exc_norm_drho_norm_drho_rho, { π , norm_drho, ρ }

exc_rho, { π , norm_drho, ρ }

exc, { π , norm_drho, ρ }

exc_norm_drho_norm_drho, { π , norm_drho, ρ }

exc_rho_rho_rho, { π , norm_drho, ρ }

exc_norm_drho_rho, { π , norm_drho, ρ }

exc_rho_rho, { π , norm_drho, ρ }

exc_norm_drho_norm_drho_norm_drho, { π , norm_drho, ρ }

```
> glob_eqs_lda4:={my_rho,my_norm_drho}union res_eqs_lda;  
glob_eqs_lda4 := {exc_norm_drho,exc_norm_drho_rho_rho,my_rho,exc_norm_drho_norm_drho_rho  
exc_rho,exc,exc_norm_drho_norm_drho,exc_rho_rho_rho,exc_norm_drho_rho,my_norm_drho  
exc_rho_rho,exc_norm_drho_norm_drho_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 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", "cg56", "cg57",  
"cg58", "cg59", "cg6", "cg60", "cg61", "cg62", "cg63", "cg64", "cg65",  
"cg66", "cg7", "cg8", "cg9"] = ["norm_drho", "k_slrho", "k_f",  
"epsilon_cGGArho", "epsilon_cGGArho", "ex_unifrho", "snorm_drho",  
"k_frhorho", "trhorho", "ex_unif", "srholrho", "tnorm_drhorho",  
"Hnorm_drhorho", "Hnorm_drho", "k_srhorho", "Arhorho", "e_c_u_0",  
"trholrho", "ex_uniflrho", "Fxrhorho", "Fxnorm_drholrho",  
"e_c_u_02rho", "Fxnorm_drhonorm_drho", "Fxrholrho", "epsilon_cGGA",  
"Fxlrho", "snorm_drholrho", "kfrhorho", "tnorm_drho", "Alrho",  
"Arholrho", "kfrhorhorho", "e_c_u_0rholrho", "tnorm_drhorhorho",  
"slrho", "e_c_u_0rho", "k_s", "trhorhorho", "rsrhorhorho",  
"tlrho", "k_f2rho", "k_s2rho", "k_srholrho", "snorm_drhorho",  
"Fxnorm_drhorho", "ex_unif2rho", "k_srho", "ex_unifrholrho",  
"ex_uniflrhorho", "ex_ldarhorhorho", "tnorm_drholrho", "k_slrho",  
"e_c_u_0lrho", "Arhorhorho", "srhorho", "Fxnorm_drho", "gamma_var",  
"e_c_u_0rhorho", "rsrhorho", "ex_unifrhorho", "k_frho", "t2norm_drho",  
"e_c_u_0lrhorho", "k_frhorhorho", "r_eqs_lda4", "e_c_u_0rhorhorho",  
"s2norm_drho", "Hnorm_drhonorm_drho"]
```

```
doubleprecision function cg66 (rho, cg, tau)  
doubleprecision exc_norm_drho_rho_rho  
doubleprecision exc_norm_drho  
doubleprecision exc_rho  
doubleprecision exc_rho_rho  
doubleprecision exc  
doubleprecision exc_norm_drho_norm_drho_rho  
doubleprecision exc_norm_drho_norm_drho
```

```
doubleprecision exc_norm_drho_rho
doubleprecision exc_rho_rho_rho
doubleprecision exc_norm_drho_norm_drho_norm_drho
doubleprecision my_rho
doubleprecision my_norm_drho
common exc_norm_drho_rho_rho, exc_norm_drho, exc_rho,
exc_rho_rho
#o, exc, exc_norm_drho_norm_drho_rho, exc_norm_drho_norm_drho,
exc_
#norm_drho_rho, exc_rho_rho_rho,
exc_norm_drho_norm_drho_norm_drho,
# my_rho, my_norm_drho
doubleprecision rho
doubleprecision cg
doubleprecision tau
doubleprecision rs2rho
doubleprecision t1021
doubleprecision t325
doubleprecision cg0
doubleprecision t738
doubleprecision t603
doubleprecision t281
doubleprecision t339
doubleprecision t133
doubleprecision t1076
doubleprecision t1885
doubleprecision t234
doubleprecision rs
doubleprecision cg1
doubleprecision t525
doubleprecision cg2
doubleprecision rsrho
doubleprecision cg3
doubleprecision t652
doubleprecision Arho
doubleprecision s
doubleprecision t1167
doubleprecision t915
doubleprecision t927
doubleprecision t573
doubleprecision cg4
doubleprecision t1720
doubleprecision cg5
doubleprecision t1865
doubleprecision t1871
doubleprecision cg6
doubleprecision t1627
```

	doubleprecision	t905
	doubleprecision	cg7
	doubleprecision	t407
	doubleprecision	cg8
	doubleprecision	t1932
	doubleprecision	t131
	doubleprecision	t132
	doubleprecision	t134
	doubleprecision	t136
	doubleprecision	t77
	doubleprecision	t79
	doubleprecision	t80
	doubleprecision	t81
	doubleprecision	t1079
	doubleprecision	t1019
	doubleprecision	t1501
	doubleprecision	t898
	doubleprecision	t1480
	doubleprecision	t962
	doubleprecision	t903
	doubleprecision	t349
	doubleprecision	t1286
	doubleprecision	cg9
	doubleprecision	t1074
	doubleprecision	cg10
	doubleprecision	cg11
	doubleprecision	cg12
	doubleprecision	cg13
	doubleprecision	cg14
	doubleprecision	Fx
	doubleprecision	t1237
	doubleprecision	cg15
	doubleprecision	t658
	doubleprecision	t1648
	doubleprecision	t1584
	doubleprecision	t432
	doubleprecision	t1080
	doubleprecision	t1081
	doubleprecision	t460
	doubleprecision	t462
	doubleprecision	t464
	doubleprecision	t467
	doubleprecision	t470
	doubleprecision	t471
	doubleprecision	t1702
	doubleprecision	Fxrho
	doubleprecision	cg16

	doubleprecision	t1433
	doubleprecision	t1443
	doubleprecision	t291
	doubleprecision	t294
	doubleprecision	cg17
	doubleprecision	t423
	doubleprecision	t145
	doubleprecision	t146
	doubleprecision	t149
	doubleprecision	t150
	doubleprecision	t152
	doubleprecision	cg18
	doubleprecision	cg19
	doubleprecision	cg20
	doubleprecision	t210
	doubleprecision	t211
	doubleprecision	t951
	doubleprecision	t664
	doubleprecision	t384
	doubleprecision	t1083
	doubleprecision	t1747
	doubleprecision	t1497
	doubleprecision	t586
	doubleprecision	t429
	doubleprecision	t63
	doubleprecision	t1636
	doubleprecision	t921
	doubleprecision	t1rho
	doubleprecision	t1330
	doubleprecision	t1333
	doubleprecision	cg21
	doubleprecision	t380
	doubleprecision	t383
	doubleprecision	t389
	doubleprecision	t392
	doubleprecision	t1086
	doubleprecision	t741
	doubleprecision	kf2rho
	doubleprecision	s2rho
	doubleprecision	t900
	doubleprecision	t1842
	doubleprecision	t960
	doubleprecision	t906
	doubleprecision	t229
	doubleprecision	t230
	doubleprecision	cg22
	doubleprecision	cg23

	doubleprecision	t282
	doubleprecision	t285
	doubleprecision	t287
	doubleprecision	t288
	doubleprecision	t418
	doubleprecision	t22
	doubleprecision	t1089
	doubleprecision	t992
	doubleprecision	t852
	doubleprecision	t618
	doubleprecision	t97
	doubleprecision	t964
	doubleprecision	t217
	doubleprecision	t218
	doubleprecision	t219
	doubleprecision	t1852
	doubleprecision	t1837
	doubleprecision	t1327
	doubleprecision	t1329
	doubleprecision	t574
	doubleprecision	t
	doubleprecision	t64
	doubleprecision	t494
	doubleprecision	t1891
	doubleprecision	t1068
	doubleprecision	t1090
	doubleprecision	t1091
	doubleprecision	t100
	doubleprecision	t101
	doubleprecision	t102
	doubleprecision	t1686
	doubleprecision	t1071
	doubleprecision	t433
	doubleprecision	t436
	doubleprecision	t437
	doubleprecision	t438
	doubleprecision	t442
	doubleprecision	t1236
	integer	t1
	integer	t2
	doubleprecision	cg24
	doubleprecision	t1786
	doubleprecision	t750
	doubleprecision	t1004
	doubleprecision	t1571
	doubleprecision	t94
	doubleprecision	cg25

	doubleprecision	t74
	doubleprecision	t75
	doubleprecision	t1094
	doubleprecision	cg26
	doubleprecision	t980
	doubleprecision	cg27
	doubleprecision	t1910
	doubleprecision	t545
	doubleprecision	t1477
	doubleprecision	t1328
	doubleprecision	mu
	doubleprecision	t194
	doubleprecision	t197
	doubleprecision	t206
	doubleprecision	t1484
	doubleprecision	t1050
	doubleprecision	t1451
	doubleprecision	t1794
	doubleprecision	t184
	doubleprecision	t1458
	doubleprecision	t1459
	doubleprecision	t1455
	doubleprecision	t943
	doubleprecision	t760
	doubleprecision	t890
	doubleprecision	t824
	doubleprecision	t1533
	doubleprecision	Alrho
	doubleprecision	cg28
	doubleprecision	t1096
	doubleprecision	t269
	doubleprecision	t273
	doubleprecision	t274
	doubleprecision	t279
	doubleprecision	cg29
	doubleprecision	t1913
	doubleprecision	t1273
	doubleprecision	t1098
	doubleprecision	kfrho
	doubleprecision	t1324
	doubleprecision	cg30
	doubleprecision	t908
	doubleprecision	t1054
	doubleprecision	t358
	doubleprecision	t84
	doubleprecision	t85
	doubleprecision	cg31

	doubleprecision	kf
	doubleprecision	t806
	doubleprecision	cg32
	doubleprecision	cg33
	doubleprecision	t293
	doubleprecision	t754
	doubleprecision	t1133
	doubleprecision	t984
	doubleprecision	t891
	doubleprecision	t1049
	doubleprecision	t1058
	doubleprecision	t553
	integer	t3
	integer	t4
	doubleprecision	t5
	doubleprecision	t1759
	doubleprecision	t1159
	doubleprecision	t311
	doubleprecision	t911
	doubleprecision	cg34
	doubleprecision	t2rho
	doubleprecision	t1924
	doubleprecision	t807
	doubleprecision	t6
	doubleprecision	t7
	doubleprecision	t8
	doubleprecision	t260
	doubleprecision	t1588
	doubleprecision	t1811
	doubleprecision	t1815
	doubleprecision	t108
	doubleprecision	t109
	doubleprecision	t110
	doubleprecision	t111
	doubleprecision	t112
	doubleprecision	t76
	doubleprecision	cg35
	doubleprecision	t57
	doubleprecision	t58
	doubleprecision	t59
	doubleprecision	t60
	doubleprecision	t559
	doubleprecision	t566
	doubleprecision	t571
	doubleprecision	cg36
	doubleprecision	t1102
	doubleprecision	cg37

	doubleprecision	t930
	doubleprecision	t190
	doubleprecision	t235
	doubleprecision	t236
	doubleprecision	t237
	doubleprecision	t240
	doubleprecision	t241
	doubleprecision	t242
	doubleprecision	t244
	doubleprecision	t245
	doubleprecision	t1514
	doubleprecision	t11
	doubleprecision	t160
	doubleprecision	t163
	doubleprecision	t164
	doubleprecision	t167
	doubleprecision	t168
	doubleprecision	t171
	doubleprecision	t172
	doubleprecision	t173
	doubleprecision	t909
	doubleprecision	t66
	doubleprecision	t1883
	doubleprecision	Fx2rho
	doubleprecision	t1517
	doubleprecision	t1520
	doubleprecision	t996
	doubleprecision	cg38
	doubleprecision	t224
	doubleprecision	t1425
	doubleprecision	t1943
	doubleprecision	t117
	doubleprecision	t120
	doubleprecision	t121
	doubleprecision	t122
	doubleprecision	t125
	doubleprecision	t129
	doubleprecision	t357
	doubleprecision	t416
	doubleprecision	srho
	doubleprecision	t318
	doubleprecision	t321
	doubleprecision	t323
	doubleprecision	t326
	doubleprecision	t329
	doubleprecision	t330
	doubleprecision	t1105

	doubleprecision	cg39
	doubleprecision	t68
	doubleprecision	t69
	doubleprecision	t12
	doubleprecision	t1166
	doubleprecision	t1552
	doubleprecision	t1240
	doubleprecision	cg40
	doubleprecision	t1578
	doubleprecision	cg41
	doubleprecision	t72
	doubleprecision	t73
	doubleprecision	t1111
	doubleprecision	A
	doubleprecision	t15
	doubleprecision	t1438
	doubleprecision	t882
	doubleprecision	t808
	doubleprecision	t1406
	doubleprecision	trho
	doubleprecision	t1087
	doubleprecision	cg42
	doubleprecision	t1053
	doubleprecision	cg43
	doubleprecision	t61
	doubleprecision	t62
	doubleprecision	Fx1rho
	doubleprecision	t153
	doubleprecision	t154
	doubleprecision	t157
	doubleprecision	t158
	doubleprecision	t159
	doubleprecision	t1806
	doubleprecision	t246
	doubleprecision	t17
	doubleprecision	t19
	doubleprecision	cg44
	doubleprecision	cg45
	doubleprecision	A2rho
	doubleprecision	cg46
	doubleprecision	cg47
	doubleprecision	cg48
	doubleprecision	t23
	doubleprecision	t324
	doubleprecision	t1446
	doubleprecision	t1951
	doubleprecision	t1956

	doubleprecision	t972
	doubleprecision	t250
	doubleprecision	t253
	doubleprecision	t254
	doubleprecision	t255
	doubleprecision	t258
	doubleprecision	t259
	doubleprecision	t261
	doubleprecision	t262
	doubleprecision	t263
	doubleprecision	t1684
	doubleprecision	t88
	doubleprecision	t89
	doubleprecision	cg49
	doubleprecision	t1176
	doubleprecision	t528
	doubleprecision	t532
	doubleprecision	t536
	doubleprecision	cg50
	doubleprecision	cg51
	doubleprecision	cg52
	doubleprecision	cg53
	doubleprecision	t576
	doubleprecision	t1112
	doubleprecision	t1172
	doubleprecision	t348
	doubleprecision	t351
	doubleprecision	t352
	doubleprecision	t354
	doubleprecision	t356
	doubleprecision	cg54
	doubleprecision	slrho
	doubleprecision	t361
	doubleprecision	t365
	doubleprecision	t366
	doubleprecision	t369
	doubleprecision	t373
	doubleprecision	t375
	doubleprecision	t376
	doubleprecision	t377
	doubleprecision	cg55
	doubleprecision	t1207
	doubleprecision	t1208
	doubleprecision	t1848
	doubleprecision	t1006
	doubleprecision	cg56
	doubleprecision	t1766

	doubleprecision	t415
	doubleprecision	t177
	doubleprecision	t178
	doubleprecision	t179
	doubleprecision	t180
	doubleprecision	t182
	doubleprecision	t186
	doubleprecision	t98
	doubleprecision	t1683
	doubleprecision	cg57
	doubleprecision	t1439
	doubleprecision	t393
	doubleprecision	t398
	doubleprecision	t402
	doubleprecision	t403
	doubleprecision	t405
	doubleprecision	t406
	doubleprecision	cg58
	doubleprecision	cg59
	doubleprecision	t956
	doubleprecision	cg60
	doubleprecision	t546
	doubleprecision	t549
	doubleprecision	t730
	doubleprecision	t1180
	doubleprecision	t612
	doubleprecision	t506
	doubleprecision	t797
	doubleprecision	t513
	doubleprecision	t518
	doubleprecision	t523
	doubleprecision	t493
	doubleprecision	t874
	doubleprecision	t500
	doubleprecision	t501
	doubleprecision	t335
	doubleprecision	t336
	doubleprecision	t342
	doubleprecision	t345
	doubleprecision	t346
	doubleprecision	cg61
	doubleprecision	cg62
	doubleprecision	t36
	doubleprecision	t1506
	doubleprecision	t1563
	doubleprecision	t1564
	doubleprecision	cg63

	doubleprecision	t686
	doubleprecision	t522
	doubleprecision	t91
	doubleprecision	t1061
	doubleprecision	t913
	doubleprecision	t453
	doubleprecision	t457
	doubleprecision	t967
	doubleprecision	cg64
	doubleprecision	cg65
	doubleprecision	t1187
	doubleprecision	t611
	doubleprecision	t1829
	doubleprecision	t1833
	doubleprecision	t49
	doubleprecision	t424
	doubleprecision	t425
	doubleprecision	t426
	doubleprecision	t428
	doubleprecision	t430
	doubleprecision	t958
	doubleprecision	t295
	doubleprecision	t299
	doubleprecision	t301
	doubleprecision	t302
	doubleprecision	t303
	doubleprecision	t304
	doubleprecision	t55
	integer	t51
	doubleprecision	t474
	doubleprecision	t477
	doubleprecision	t478
	doubleprecision	t479
	doubleprecision	t482
	doubleprecision	t486
	doubleprecision	t490
	doubleprecision	t1442
	doubleprecision	t914
	doubleprecision	t876
	doubleprecision	t878
	doubleprecision	t879
	doubleprecision	t877
	doubleprecision	t1100
	doubleprecision	t1876
	doubleprecision	t628
	doubleprecision	t629
	doubleprecision	t634

```

doubleprecision t138
doubleprecision t140
doubleprecision t141
doubleprecision t142
doubleprecision t1130
doubleprecision t1194
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 = 0.1D1 / 0.3141592654D1
t6 = 0.1D1 / rho
t7 = t5 * t6
t8 = t7 ** (0.1D1 / 0.3D1)
rs = dble(t4) * t8 / 0.4D1
t11 = 0.1D1 + 0.21370D0 * rs
t12 = sqrt(rs)
t15 = t12 * rs
t17 = rs ** 0.20D1
t19 = 0.75957D1 * t12 + 0.35876D1 * rs + 0.16382D1 * t15 +
0.492
#94D0 * t17
t22 = 0.1D1 + 0.1608182432D2 / t19
t23 = log(t22)
cg22 = -0.62182D-1 * t11 * t23
t36 = log(0.1D1 + 0.3216468318D2 / (0.141189D2 * t12 +
0.61977D1
# * rs + 0.33662D1 * t15 + 0.62517D0 * t17))
t49 = log(0.1D1 + 0.2960857464D1 / (0.10357D2 * t12 + 0.36231D1
#* rs + 0.88026D0 * t15 + 0.49671D0 * t17))
t51 = 2 ** (0.1D1 / 0.3D1)
t55 = log(0.2D1)
t57 = 0.3141592654D1 ** 2
t58 = 0.1D1 / t57
cg59 = (0.1D1 - t55) * t58
t59 = t57 * rho
t60 = t59 ** (0.1D1 / 0.3D1)
cg1 = dble(t1) * t60
t61 = cg1 * t5
t62 = sqrt(t61)
cg40 = 0.2D1 * t62
t63 = 0.1D1 / cg40
t64 = cg * t63
t = t64 * t6 / 0.2D1
t66 = 0.1D1 / cg59

```

```

t68 = exp(-cg22 * t66)
t69 = -0.1D1 + t68
A = 0.66725D-1 * t66 / t69
t72 = t ** 2
t73 = t66 * t72
t74 = A * t72
t75 = 0.1D1 + t74
t76 = A ** 2
t77 = t72 ** 2
t79 = 0.1D1 + t74 + t76 * t77
t80 = 0.1D1 / t79
t81 = t75 * t80
t84 = 0.1D1 + 0.66725D-1 * t73 * t81
t85 = log(t84)
cg3 = cg22 + cg59 * t85
mu = 0.2224166667D-1 * t57
kf = cg1
cg16 = -0.3D1 / 0.4D1 * t5 * kf
t88 = 0.1D1 / kf
t89 = cg * t88
s = t89 * t6 / 0.2D1
t91 = s ** 2
t94 = 0.1D1 + 0.1243781095D1 * mu * t91
Fx = 0.1804D1 - 0.804D0 / t94
t97 = rho * cg16
exc = t97 * Fx + rho * cg3
t98 = t8 ** 2
t100 = 0.1D1 / t98 * t5
t101 = rho ** 2
t102 = 0.1D1 / t101
rsrho = -dble(t4) * t100 * t102 / 0.12D2
t108 = t19 ** 2
t109 = 0.1D1 / t108
t110 = t11 * t109
t111 = 0.1D1 / t12
t112 = t111 * rsrho
t117 = rs ** 0.10D1
t120 = 0.3797850000D1 * t112 + 0.35876D1 * rsrho +
0.2457300000D
#1 * t12 * rsrho + 0.985880D0 * t117 * rsrho
t121 = 0.1D1 / t22
t122 = t120 * t121
cg4 = -0.1328829340D-1 * rsrho * t23 + 0.9999999999D0 * t110 *
t
#122
t125 = t60 ** 2
cg62 = dble(t1) / t125 * t57 / 0.3D1

```

```

t129 = 0.1D1 / t62
cg5 = t129 * cg62 * t5
t131 = cg40 ** 2
t132 = 0.1D1 / t131
t133 = cg * t132
t134 = t6 * cg5
t136 = t64 * t102
trho = -t133 * t134 / 0.2D1 - t136 / 0.2D1
t138 = cg59 ** 2
t140 = t69 ** 2
t141 = 0.1D1 / t140
t142 = 0.1D1 / t138 * t141
Arho = 0.66725D-1 * t142 * cg4 * t68
t145 = t66 * t
t146 = t81 * trho
t149 = Arho * t72
t150 = A * t
t152 = 0.2D1 * t150 * trho
t153 = t149 + t152
t154 = t153 * t80
t157 = t79 ** 2
t158 = 0.1D1 / t157
t159 = t75 * t158
t160 = A * t77
t163 = t72 * t
t164 = t76 * t163
t167 = t149 + t152 + 0.2D1 * t160 * Arho + 0.4D1 * t164 * trho
t168 = t159 * t167
t171 = 0.133450D0 * t145 * t146 + 0.66725D-1 * t73 * t154 -
0.66
#725D-1 * t73 * t168
t172 = cg59 * t171
t173 = 0.1D1 / t84
cg11 = cg4 + t172 * t173
kfrho = cg62
cg12 = -0.3D1 / 0.4D1 * t5 * kfrho
t177 = kf ** 2
t178 = 0.1D1 / t177
t179 = cg * t178
t180 = t6 * kfrho
t182 = t89 * t102
srho = -t179 * t180 / 0.2D1 - t182 / 0.2D1
t184 = t94 ** 2
t186 = 0.1D1 / t184 * mu
Fxrho = 0.2000000001D1 * t186 * s * srho
t190 = rho * cg12
exc_rho = cg16 * Fx + t190 * Fx + t97 * Fxrho + cg3 + rho *

```

cg11

```
cg33 = t63 * t6 / 0.2D1
t194 = t81 * cg33
t197 = t66 * t163
t206 = 0.2D1 * t150 * cg33 + 0.4D1 * t164 * cg33
t210 = 0.133450D0 * t145 * t194 + 0.133450D0 * t197 * A * cg33
```

*

```
# t80 - 0.66725D-1 * t73 * t159 * t206
t211 = cg59 * t210
cg2 = t211 * t173
cg13 = t88 * t6 / 0.2D1
cg58 = 0.2000000001D1 * t186 * s * cg13
exc_norm_drho = t97 * cg58 + rho * cg2
t217 = 0.1D1 / t98 / t7 * t58
t218 = t101 ** 2
t219 = 0.1D1 / t218
t224 = 0.1D1 / t101 / rho
cg60 = -dble(t4) * t217 * t219 / 0.18D2 + dble(t4) * t100 *
```

t224

```
# / 0.6D1
t229 = 0.1328829340D-1 * cg60 * t23
t230 = rsrho * t109
t234 = 0.1D1 / t108 / t19
t235 = t11 * t234
t236 = t120 ** 2
t237 = t236 * t121
t240 = 0.1D1 / t15
t241 = rsrho ** 2
t242 = t240 * t241
t244 = t111 * cg60
t245 = 0.3797850000D1 * t244
t246 = 0.35876D1 * cg60
t250 = 0.2457300000D1 * t12 * cg60
t253 = 0.985880D0 * t117 * cg60
t254 = -0.1898925000D1 * t242 + t245 + t246 + 0.1228650000D1 *
```

t

```
#111 * t241 + t250 + 0.9858800D0 * t241 + t253
t255 = t254 * t121
t258 = t108 ** 2
t259 = 0.1D1 / t258
t260 = t11 * t259
t261 = t22 ** 2
t262 = 0.1D1 / t261
t263 = t236 * t262
cg6 = -t229 + 0.4274000000D0 * t230 * t122 - 0.2000000000D1 *
```

t2

```
#35 * t237 + 0.9999999999D0 * t110 * t255 + 0.1608182432D2 * t260
```

*

t263

```
cg55 = cg4
t269 = t57 ** 2
cg14 = -0.2D1 / 0.9D1 * dble(t1) / t125 / t59 * t269
t273 = 0.1D1 / t62 / t61
t274 = cg62 ** 2
t279 = t129 * cg14 * t5
cg20 = -t273 * t274 * t58 / 0.2D1 + t279
cg0 = cg5
t281 = 0.1D1 / t131 / cg40
t282 = cg * t281
t285 = t102 * cg5
t287 = t133 * t285 / 0.2D1
t288 = t6 * cg20
t291 = t102 * cg0
t293 = t133 * t291 / 0.2D1
t294 = t64 * t224
cg15 = t282 * t134 * cg0 + t287 - t133 * t288 / 0.2D1 + t293 +
```

t

#294

```
t295 = t6 * cg0
t1rho = -t133 * t295 / 0.2D1 - t136 / 0.2D1
t299 = 0.1D1 / t138 / cg59
t301 = 0.1D1 / t140 / t69
t302 = t299 * t301
t303 = t68 ** 2
t304 = cg4 * t303
t311 = t299 * t141
cg21 = 0.133450D0 * t302 * t304 * cg55 + 0.66725D-1 * t142 *
```

cg6

* t68 - 0.66725D-1 * t311 * cg4 * cg55 * t68

```
Alrho = 0.66725D-1 * t142 * cg55 * t68
t318 = t66 * t1rho
t321 = Alrho * t72
t323 = 0.2D1 * t150 * t1rho
t324 = t321 + t323
t325 = t324 * t80
t326 = t325 * trho
t329 = t145 * t75
t330 = t158 * trho
t335 = t321 + t323 + 0.2D1 * t160 * Alrho + 0.4D1 * t164 *
```

t1rho

```
t336 = t330 * t335
t339 = t81 * cg15
t342 = t154 * t1rho
t345 = cg21 * t72
```

```

t346 = Arho * t
t348 = 0.2D1 * t346 * t1rho
t349 = Alrho * t
t351 = 0.2D1 * t349 * trho
t352 = A * t1rho
t354 = 0.2D1 * t352 * trho
t356 = 0.2D1 * t150 * cg15
t357 = t345 + t348 + t351 + t354 + t356
t358 = t357 * t80
t361 = t153 * t158
t365 = t158 * t167
t366 = t365 * t1rho
t369 = t324 * t158
t373 = t73 * t75
t375 = 0.1D1 / t157 / t79
t376 = t375 * t167
t377 = t376 * t335
t380 = Alrho * t77
t383 = A * t163
t384 = Arho * t1rho
t389 = trho * Alrho
t392 = t76 * t72
t393 = trho * t1rho
t398 = t345 + t348 + t351 + t354 + t356 + 0.2D1 * t380 * Arho +
#0.8D1 * t383 * t384 + 0.2D1 * t160 * cg21 + 0.8D1 * t383 * t389 +
#0.12D2 * t392 * t393 + 0.4D1 * t164 * cg15
t402 = 0.133450D0 * t318 * t146 + 0.133450D0 * t145 * t326 -
0.1
#33450D0 * t329 * t336 + 0.133450D0 * t145 * t339 + 0.133450D0 *
t1
#45 * t342 + 0.66725D-1 * t73 * t358 - 0.66725D-1 * t73 * t361 *
t3
#35 - 0.133450D0 * t329 * t366 - 0.66725D-1 * t73 * t369 * t167 +
0
#.133450D0 * t373 * t377 - 0.66725D-1 * t73 * t159 * t398
t403 = cg59 * t402
t405 = t84 ** 2
t406 = 0.1D1 / t405
t407 = t81 * t1rho
t415 = 0.133450D0 * t145 * t407 + 0.66725D-1 * t73 * t325 -
0.66
#725D-1 * t73 * t159 * t335
t416 = t406 * t415
t418 = cg59 * t415
cg10 = cg6 + t403 * t173 - t172 * t416
cg32 = cg14
cg61 = -0.3D1 / 0.4D1 * t5 * cg32

```

```

cg24 = cg12
t423 = 0.1D1 / t177 / kf
t424 = cg * t423
t425 = kfrho ** 2
t426 = t6 * t425
t428 = t102 * kfrho
t429 = t179 * t428
t430 = t6 * cg32
t432 = t179 * t430 / 0.2D1
t433 = t89 * t224
cg57 = t424 * t426 + t429 - t432 + t433
slrho = srho
t436 = mu ** 2
t437 = 0.1D1 / t184 / t94 * t436
t438 = t91 * srho
t442 = slrho * srho
cg25 = -0.9950248765D1 * t437 * t438 * slrho + 0.2000000001D1 *
#t186 * t442 + 0.2000000001D1 * t186 * s * cg57
Fxlrho = 0.2000000001D1 * t186 * s * slrho
t453 = rho * cg61
t457 = rho * cg24
exc_rho_rho = cg24 * Fx + cg16 * Fxlrho + cg12 * Fx + t453 * Fx
#+ t190 * Fxlrho + cg16 * Fxrho + t457 * Fxrho + t97 * cg25 + cg55
#+ t418 * t173 + cg11 + rho * cg10
t460 = t132 * t6
t462 = t63 * t102
cg18 = -t460 * cg5 / 0.2D1 - t462 / 0.2D1
t464 = t66 * trho
t467 = t154 * cg33
t470 = t158 * cg33
t471 = t470 * t167
t474 = t81 * cg18
t477 = t73 * A
t478 = cg33 * t80
t479 = t478 * trho
t482 = Arho * cg33
t486 = A * cg18
t490 = t197 * A
t493 = t158 * t206
t494 = t493 * trho
t500 = t375 * t206
t501 = t500 * t167
t506 = A * trho
t513 = cg33 * trho
t518 = 0.2D1 * t346 * cg33 + 0.2D1 * t506 * cg33 + 0.2D1 * t150
#* cg18 + 0.8D1 * t383 * t482 + 0.12D2 * t392 * t513 + 0.4D1 *

```

t164


```

# * cg18
t522 = 0.133450D0 * t464 * t194 + 0.133450D0 * t145 * t467 -
0.1
#33450D0 * t329 * t471 + 0.133450D0 * t145 * t474 + 0.400350D0 *
t4
#77 * t479 + 0.133450D0 * t197 * t482 * t80 + 0.133450D0 * t197 *
t
#486 * t80 - 0.133450D0 * t490 * t471 - 0.133450D0 * t329 * t494 -
#0.66725D-1 * t73 * t361 * t206 + 0.133450D0 * t373 * t501 -
0.6672
#5D-1 * t73 * t159 * t518
t523 = cg59 * t522
t525 = t406 * t171
cg19 = t523 * t173 - t211 * t525
t528 = t178 * t6
cg47 = -t528 * kfrho / 0.2D1 - t88 * t102 / 0.2D1
t532 = t91 * cg13
t536 = srho * cg13
cg48 = -0.9950248765D1 * t437 * t532 * srho + 0.2000000001D1 *
t
#186 * t536 + 0.2000000001D1 * t186 * s * cg47
exc_norm_drho_rho = cg16 * cg58 + t190 * cg58 + t97 * cg48 +
cg2
# + rho * cg19
t545 = cg33 ** 2
t546 = t66 * t545
t549 = A * t545
t553 = t470 * t206
t559 = t206 ** 2
t566 = 0.2D1 * t549 + 0.12D2 * t392 * t545
t571 = cg59 * (0.133450D0 * t546 * t81 + 0.667250D0 * t73 *
t549
# * t80 - 0.266900D0 * t329 * t553 - 0.266900D0 * t490 * t553 +
0.1
#33450D0 * t73 * t75 * t375 * t559 - 0.66725D-1 * t73 * t159 *
t566
#)
t573 = t210 ** 2
t574 = cg59 * t573
cg9 = t571 * t173 - t574 * t406
t576 = cg13 ** 2
cg28 = -0.9950248765D1 * t437 * t91 * t576 + 0.2000000001D1 *
t1
#86 * t576
exc_norm_drho_norm_drho = t97 * cg28 + rho * cg9
t586 = 0.1D1 / t57 / 0.3141592654D1
cg42 = -0.5D1 / 0.54D2 * dble(t4) / t98 / t58 / t102 * t586 /

```

```

t2
#18 / t101 + dble(t4) * t217 / t218 / rho / 0.3D1 - dble(t4) *
t100
# * t219 / 0.2D1
rs2rho = rsrho
t603 = cg60 * t109
t611 = 0.3797850000D1 * t111 * rs2rho + 0.35876D1 * rs2rho +
0.2
#4573000000D1 * t12 * rs2rho + 0.985880D0 * t117 * rs2rho
t612 = t611 * t121
t618 = t122 * t611
t628 = -0.1898925000D1 * t240 * rsrho * rs2rho + t245 + t246 +
0
#.1228650000D1 * t112 * rs2rho + t250 + 0.9858800D0 * rsrho *
rs2rh
#o + t253
t629 = t628 * t121
t634 = t120 * t262 * t611
t652 = rs2rho * t109
t658 = rs ** 2
t664 = t240 * cg60
t686 = 0.2848387500D1 / t12 / t658 * t241 * rs2rho -
0.379785000
#0D1 * t664 * rsrho - 0.1898925000D1 * t664 * rs2rho +
0.3797850000
#D1 * t111 * cg42 + 0.35876D1 * cg42 - 0.6143250000D0 * t242 *
rs2r
#ho + 0.2457300000D1 * t244 * rsrho + 0.1228650000D1 * t244 *
rs2rh
#o + 0.2457300000D1 * t12 * cg42 + 0.19717600D1 * cg60 * rsrho +
0.
#9858800D0 * cg60 * rs2rho + 0.985880D0 * t117 * cg42
cg7 = -0.1328829340D-1 * cg42 * t23 + 0.2137000000D0 * t603 *
t6
#12 + 0.4274000000D0 * t603 * t122 - 0.8548000000D0 * rsrho * t234
#* t618 + 0.4274000000D0 * t230 * t629 + 0.6873371714D1 * rsrho *
t
#259 * t634 - 0.4274000000D0 * rs2rho * t234 * t237 +
0.6000000000D
#1 * t260 * t237 * t611 - 0.4000000000D1 * t235 * t629 * t120 -
0.9
#649094592D2 * t11 / t258 / t19 * t263 * t611 + 0.2137000000D0 *
t6
#52 * t255 - 0.2000000000D1 * t235 * t255 * t611 + 0.9999999999D0
*
# t110 * t686 * t121 + 0.1608182432D2 * t260 * t254 * t262 * t611
+

```

```

# 0.3436685857D1 * rs2rho * t259 * t263 + 0.3216364864D2 * t260 *
t
#628 * t262 * t120 + 0.5172501469D3 * t11 / t258 / t108 * t236 /
t2
#61 / t22 * t611
cg37 = -t229 + 0.2137000000D0 * t230 * t612 + 0.2137000000D0 *
t
#652 * t122 - 0.2000000000D1 * t235 * t618 + 0.9999999999D0 * t110
#* t629 + 0.1608182432D2 * t260 * t634
cg64 = cg37
cg27 = -0.1328829340D-1 * rs2rho * t23 + 0.9999999999D0 * t110
*
# t612
cg65 = 0.10D2 / 0.27D2 * db1e(t1) / t125 / t101 * t57
cg44 = kfrho
t730 = cg1 ** 2
t738 = t273 * cg14
t741 = t58 * cg44
cg46 = -t273 * cg62 * t741 / 0.2D1 + t279
cg54 = cg46
cg45 = t129 * cg44 * t5
t750 = t131 ** 2
t754 = cg5 * cg0
t760 = t6 * cg46
cg41 = -0.3D1 * cg / t750 * t6 * t754 * cg45 - t282 * t285 *
cg0
# + t282 * t760 * cg0 + t282 * t134 * cg54 - t282 * t285 * cg45 -
t
#133 * t224 * cg5 + t133 * t102 * cg46 / 0.2D1 + t282 * t288 *
cg45
# + t133 * t102 * cg20 / 0.2D1 - t133 * t6 * (0.3D1 / 0.4D1 / t62
/
# t730 / t58 * t274 * t586 * cg44 - t738 * t58 * cg62 - t738 *
t741
# / 0.2D1 + t129 * cg65 * t5) / 0.2D1 - t282 * t291 * cg45 - t133
*
# t224 * cg0 + t133 * t102 * cg54 / 0.2D1 - t133 * t224 * cg45 -
0.
#3D1 * t64 * t219
t797 = t133 * t102 * cg45 / 0.2D1
cg23 = t282 * t134 * cg45 + t287 - t133 * t760 / 0.2D1 + t797 +
#t294
cg43 = t282 * t295 * cg45 + t293 - t133 * t6 * cg54 / 0.2D1 +
t7
#97 + t294
t2rho = -t133 * t6 * cg45 / 0.2D1 - t136 / 0.2D1
t806 = t138 ** 2

```

```

t807 = 0.1D1 / t806
t808 = t140 ** 2
t824 = t303 * cg55 * cg27
t852 = cg55 * cg27 * t68
cg56 = 0.400350D0 * t807 / t808 * cg4 * t303 * t68 * cg55 *
cg27
# + 0.133450D0 * t302 * cg37 * t303 * cg55 - 0.400350D0 * t807 *
t3
#01 * cg4 * t824 + 0.133450D0 * t302 * t304 * cg64 + 0.133450D0 *
t
#302 * cg6 * t303 * cg27 + 0.66725D-1 * t142 * cg7 * t68 -
0.66725D
#-1 * t311 * cg6 * cg27 * t68 - 0.66725D-1 * t311 * cg37 * cg55 *
t
#68 - 0.66725D-1 * t311 * cg4 * cg64 * t68 + 0.66725D-1 * t807 *
t1
#41 * cg4 * t852
cg35 = 0.133450D0 * t302 * t304 * cg27 + 0.66725D-1 * t142 *
cg3
#7 * t68 - 0.66725D-1 * t311 * cg4 * cg27 * t68
t68
cg34 = 0.133450D0 * t302 * t824 + 0.66725D-1 * t142 * cg64 *
# - 0.66725D-1 * t311 * t852
A2rho = 0.66725D-1 * t142 * cg27 * t68
t874 = A2rho * t72
t876 = 0.2D1 * t150 * t2rho
t877 = t874 + t876
t878 = t877 * t80
t879 = t878 * trho
t882 = t66 * t2rho
t890 = t874 + t876 + 0.2D1 * t160 * A2rho + 0.4D1 * t164 *
t2rho
t891 = t335 * t890
t898 = t73 * t153
t900 = t375 * t335 * t890
t903 = cg35 * t72
t905 = 0.2D1 * t346 * t2rho
t906 = A2rho * t
t908 = 0.2D1 * t906 * trho
t909 = A * t2rho
t911 = 0.2D1 * t909 * trho
t913 = 0.2D1 * t150 * cg23
t914 = t903 + t905 + t908 + t911 + t913
t915 = t914 * t80
t921 = t882 * t75
t927 = A2rho * t77
t930 = Arho * t2rho

```

```

t943 = t903 + t905 + t908 + t911 + t913 + 0.2D1 * t927 * Arho +
#0.8D1 * t383 * t930 + 0.2D1 * t160 * cg35 + 0.8D1 * t383 * trho *
#A2rho + 0.12D2 * t392 * trho * t2rho + 0.4D1 * t164 * cg23
t951 = 0.133450D0 * t318 * t879 + 0.133450D0 * t882 * t339 +
0.2
#66900D0 * t329 * t375 * trho * t891 + 0.133450D0 * t145 * t325 *
c
#g23 + 0.133450D0 * t898 * t900 + 0.133450D0 * t145 * t915 * t1rho
#+ 0.133450D0 * t882 * t326 - 0.133450D0 * t921 * t336 +
0.133450D0
# * t145 * t154 * cg43 + 0.133450D0 * t373 * t375 * t943 * t335 +
0
#.133450D0 * t73 * t877 * t377
t956 = cg34 * t72
t958 = 0.2D1 * t349 * t2rho
t960 = 0.2D1 * t906 * t1rho
t962 = 0.2D1 * t909 * t1rho
t964 = 0.2D1 * t150 * cg43
t967 = Alrho * t2rho
t972 = t1rho * A2rho
t980 = t956 + t958 + t960 + t962 + t964 + 0.2D1 * t927 * Alrho
+
# 0.8D1 * t383 * t967 + 0.2D1 * t160 * cg34 + 0.8D1 * t383 * t972
+
# 0.12D2 * t392 * t1rho * t2rho + 0.4D1 * t164 * cg43
t984 = t956 + t958 + t960 + t962 + t964
t992 = t357 * t158
t996 = t145 * t877
t1004 = t145 * t153
t1006 = t158 * t335 * t2rho
t1019 = -0.133450D0 * t329 * t158 * cg23 * t335 - 0.66725D-1 *
t
#73 * t361 * t980 - 0.66725D-1 * t73 * t984 * t158 * t167 +
0.13345
#0D0 * t145 * t358 * t2rho - 0.66725D-1 * t73 * t992 * t890 -
0.133
#450D0 * t996 * t336 - 0.133450D0 * t996 * t366 - 0.133450D0 *
t329
# * t330 * t980 - 0.133450D0 * t1004 * t1006 + 0.266900D0 * t329 *
#t376 * t1rho * t890 - 0.133450D0 * t329 * t365 * cg43 -
0.66725D-1
# * t73 * t369 * t943
t1021 = t984 * t80
t1049 = t73 * t324
t1050 = t376 * t890
t1053 = t318 * t75
t1054 = t330 * t890

```

```

t1058 = t158 * t1rho * t890
t1061 = 0.133450D0 * t145 * t1021 * trho - 0.133450D0 * t921 *
t
#366 + 0.266900D0 * t329 * t376 * t335 * t2rho + 0.133450D0 * t373
#* t375 * t398 * t890 - 0.66725D-1 * t73 * t914 * t158 * t335 +
0.1
#33450D0 * t145 * t878 * cg15 - 0.133450D0 * t329 * t158 * cg15 *
t
#890 + 0.133450D0 * t66 * cg43 * t146 + 0.133450D0 * t1049 * t1050
#- 0.133450D0 * t1053 * t1054 - 0.133450D0 * t1004 * t1058
t1068 = t145 * t324
t1071 = t967 * trho
t1074 = cg21 * t
t1076 = 0.2D1 * t1074 * t2rho
t1079 = 0.2D1 * cg35 * t * t1rho
t1080 = t930 * t1rho
t1081 = 0.2D1 * t1080
t1083 = 0.2D1 * t346 * cg43
t1086 = 0.2D1 * cg34 * t * trho
t1087 = 0.2D1 * t1071
t1089 = 0.2D1 * t349 * cg23
t1090 = t972 * trho
t1091 = 0.2D1 * t1090
t1094 = 0.2D1 * A * cg43 * trho
t1096 = 0.2D1 * t352 * cg23
t1098 = 0.2D1 * t906 * cg15
t1100 = 0.2D1 * t909 * cg15
t1102 = 0.2D1 * t150 * cg41
t1105 = t76 * t
t1111 = 0.24D2 * t74 * t1071 + t1076 + t1079 + t1081 + t1083 +
t
#1086 + t1087 + t1089 + t1091 + t1094 + t1096 + t1098 + t1100 +
t11
#02 + 0.24D2 * t74 * t1090 + 0.24D2 * t1105 * t393 * t2rho +
0.24D2
# * t74 * t1080
t1112 = cg56 * t72
t1130 = A2rho * t163
t1133 = A1rho * t163
t1159 = t1112 + 0.2D1 * t380 * cg35 + 0.2D1 * t927 * cg21 +
0.4D
#1 * t164 * cg41 + 0.2D1 * t160 * cg56 + 0.8D1 * t383 * trho *
cg34
# + 0.12D2 * t392 * trho * cg43 + 0.8D1 * t383 * cg23 * A1rho +
0.8
#D1 * t1130 * t389 + 0.8D1 * t1133 * t930 + 0.12D2 * t392 * cg23 *
#t1rho + 0.8D1 * t383 * cg35 * t1rho + 0.2D1 * cg34 * t77 * Arho +

```

```

#0.8D1 * t1130 * t384 + 0.12D2 * t392 * cg15 * t2rho + 0.8D1 *
t383
# * Arho * cg43 + 0.8D1 * t383 * cg21 * t2rho + 0.8D1 * t383 *
cg15
# * A2rho
    t1166 = t157 ** 2
    t1167 = 0.1D1 / t1166
    t1172 = t877 * t158
    t1176 = t1112 + t1076 + t1079 + t1081 + t1083 + t1086 + t1087 +
#t1089 + t1091 + t1094 + t1096 + t1098 + t1100 + t1102
    t1180 = t365 * t2rho
    t1187 = t81 * cg23
    t1194 = 0.133450D0 * t145 * t81 * cg41 + 0.133450D0 * t373 *
t37
#6 * t980 - 0.133450D0 * t1068 * t1054 - 0.66725D-1 * t73 * t159 *
#(t1111 + t1159) + 0.133450D0 * t882 * t342 - 0.400350D0 * t373 *
t
#1167 * t167 * t891 - 0.66725D-1 * t73 * t1172 * t398 + 0.66725D-1
#* t73 * t1176 * t80 - 0.133450D0 * t1068 * t1180 - 0.133450D0 *
t3
#29 * t158 * t398 * t2rho + 0.133450D0 * t318 * t1187 - 0.133450D0
#* t329 * t158 * t943 * t1rho
    t1207 = 0.133450D0 * t145 * t81 * t2rho + 0.66725D-1 * t73 *
t87
#8 - 0.66725D-1 * t73 * t159 * t890
    t1208 = t406 * t1207
    t1236 = 0.133450D0 * t882 * t146 + 0.133450D0 * t145 * t879 -
0.
#133450D0 * t329 * t1054 + 0.133450D0 * t145 * t1187 + 0.133450D0
*
# t145 * t154 * t2rho + 0.66725D-1 * t73 * t915 - 0.66725D-1 * t73
#* t361 * t890 - 0.133450D0 * t329 * t1180 - 0.66725D-1 * t73 *
t11
#72 * t167 + 0.133450D0 * t373 * t1050 - 0.66725D-1 * t73 * t159 *
#t943
    t1237 = cg59 * t1236
    t1240 = 0.1D1 / t405 / t84
    t1273 = 0.133450D0 * t882 * t407 + 0.133450D0 * t145 * t878 *
t1
#rho - 0.133450D0 * t329 * t1058 + 0.133450D0 * t145 * t81 * cg43
+
# 0.133450D0 * t145 * t325 * t2rho + 0.66725D-1 * t73 * t1021 -
0.6
#6725D-1 * t73 * t369 * t890 - 0.133450D0 * t329 * t1006 -
0.66725D
#-1 * t73 * t1172 * t335 + 0.133450D0 * t373 * t900 - 0.66725D-1 *
#t73 * t159 * t980

```

```

cg36 = cg65
kf2rho = cg44
cg50 = cg61
cg51 = cg50
cg49 = -0.3D1 / 0.4D1 * t5 * kf2rho
t1286 = t177 ** 2
cg17 = t424 * t180 * kf2rho + t429 / 0.2D1 - t432 + t179 * t102
#* kf2rho / 0.2D1 + t433
cg39 = cg17
s2rho = -t179 * t6 * kf2rho / 0.2D1 - t182 / 0.2D1
t1324 = t184 ** 2
t1327 = 0.1D1 / t1324 * t436 * mu
t1328 = t91 * s
t1329 = t1327 * t1328
t1330 = t442 * s2rho
t1333 = t437 * s
cg29 = -0.9950248765D1 * t437 * t438 * s2rho + 0.2000000001D1 *
#t186 * s2rho * srho + 0.2000000001D1 * t186 * s * cg17
cg30 = -0.9950248765D1 * t437 * t91 * slrho * s2rho +
0.20000000
#01D1 * t186 * s2rho * slrho + 0.2000000001D1 * t186 * s * cg39
Fx2rho = 0.2000000001D1 * t186 * s * s2rho
cg52 = cg51 * Fx + cg24 * Fx2rho + cg49 * Fx1rho + cg16 * cg30
+
# cg50 * Fx + cg12 * Fx2rho + cg61 * Fx - 0.3D1 / 0.4D1 * rho * t5
#* cg36 * Fx + t453 * Fx2rho + cg12 * Fx1rho + rho * cg50 * Fx1rho
#+ t190 * cg30 + cg49 * Fxrho + cg16 * cg29 + cg24 * Fxrho + rho *
#cg51 * Fxrho + t457 * cg29 + cg16 * cg25 + rho * cg49 * cg25 +
t97
# * (0.7425558783D2 * t1329 * t1330 - 0.2985074630D2 * t1333 *
t133
#0 - 0.9950248765D1 * t437 * t91 * cg17 * slrho - 0.9950248765D1 *
#t437 * t438 * cg39 + 0.2000000001D1 * t186 * cg39 * srho +
0.20000
#00001D1 * t186 * slrho * cg17 - 0.9950248765D1 * t437 * t91 *
cg57
# * s2rho + 0.2000000001D1 * t186 * s2rho * cg57 + 0.2000000001D1
*
# t186 * s * (-0.3D1 * cg / t1286 * t426 * kf2rho - t424 * t102 *
t
#425 + 0.2D1 * t424 * t430 * kfrho - 0.2D1 * t424 * t428 * kf2rho
-
# 0.2D1 * t179 * t224 * kfrho + 0.3D1 / 0.2D1 * t179 * t102 * cg32
#+ t424 * t430 * kf2rho - t179 * t6 * cg36 / 0.2D1 - t179 * t224 *
#kf2rho - 0.3D1 * t89 * t219))
exc_rho_rho_rho = cg52 + cg64 + cg59 * t1273 * t173 - t418 *
t12

```



```

#08 + cg37 + t1237 * t173 - t172 * t1208 + cg10 + rho * (cg7 +
cg59
# * (t951 + t1019 + t1061 + t1194) * t173 - t403 * t1208 - t1237 *
#t416 + 0.2D1 * t172 * t1240 * t415 * t1207 - t172 * t406 * t1273)
t1406 = t132 * t102
cg38 = t281 * t6 * t754 + t1406 * cg5 / 0.2D1 - t460 * cg20 /
0.
#2D1 + t1406 * cg0 / 0.2D1 + t63 * t224
cg53 = -t460 * cg0 / 0.2D1 - t462 / 0.2D1
t1425 = cg33 * Alrho
t1433 = 0.2D1 * t349 * cg33 + 0.2D1 * t352 * cg33 + 0.2D1 *
t150
# * cg53 + 0.8D1 * t383 * t1425 + 0.12D2 * t392 * cg33 * t1rho +
0.
#4D1 * t164 * cg53
t1438 = t197 * Arho
t1439 = t470 * t335
t1442 = cg18 * t158
t1443 = t1442 * t335
t1446 = Arho * cg53
t1451 = t167 * t335
t1455 = t464 * t75
t1458 = t375 * cg33
t1459 = t1458 * t1451
t1477 = 0.133450D0 * t66 * cg15 * t194 + 0.133450D0 * t373 *
t37
#5 * t1433 * t167 - 0.133450D0 * t1438 * t1439 - 0.133450D0 * t490
#* t1443 + 0.133450D0 * t197 * t1446 * t80 - 0.400350D0 * t373 *
t1
#167 * t206 * t1451 - 0.133450D0 * t1455 * t1439 + 0.266900D0 *
t32
#9 * t1459 - 0.66725D-1 * t73 * t369 * t518 + 0.133450D0 * t373 *
t
#500 * t398 + 0.133450D0 * t145 * t325 * cg18 + 0.133450D0 * t318
*
# t474 - 0.133450D0 * t329 * t158 * t518 * t1rho
t1480 = t500 * t335
t1484 = t478 * t1rho
t1497 = trho * t335
t1501 = t81 * cg53
t1506 = t145 * A
t1514 = t158 * cg53 * t167
t1517 = t470 * t398
t1520 = -0.133450D0 * t329 * t1443 + 0.133450D0 * t898 * t1480
+
# 0.400350D0 * t73 * Arho * t1484 - 0.66725D-1 * t73 * t361 *
t1433

```

```

# + 0.400350D0 * t477 * cg18 * t80 * tlrho + 0.133450D0 * t145 *
t3
#58 * cg33 - 0.400350D0 * t477 * t470 * t1497 + 0.133450D0 * t464
*
# t1501 - 0.133450D0 * t1068 * t494 + 0.800700D0 * t1506 * t478 *
t
#393 + 0.400350D0 * t73 * Alrho * t479 - 0.133450D0 * t329 * t1514
#- 0.133450D0 * t490 * t1517
    t1533 = t375 * t518
    t1552 = cg21 * cg33
    t1563 = 0.133450D0 * t197 * A * cg38 * t80 - 0.133450D0 * t197
*
    # Alrho * t471 + 0.400350D0 * t477 * cg53 * t80 * trho +
0.133450D0
    # * t373 * t1533 * t335 - 0.133450D0 * t490 * t1514 - 0.133450D0 *
#t1004 * t1439 + 0.133450D0 * t145 * t81 * cg38 + 0.266900D0 *
t490
    # * t1459 + 0.133450D0 * t145 * t154 * cg53 + 0.400350D0 * t477 *
t
    #478 * cg15 + 0.133450D0 * t197 * t1552 * t80 - 0.133450D0 * t329
*
    # t158 * t1433 * trho - 0.133450D0 * t329 * t493 * cg15
        t1564 = t167 * tlrho
        t1571 = t325 * cg33
        t1578 = Alrho * cg18
        t1584 = t384 * cg33
        t1588 = t389 * cg33
        t1627 = 0.2D1 * t1074 * cg33 + 0.2D1 * t1584 + 0.2D1 * t346 *
cg
    #53 + 0.2D1 * t1588 + 0.2D1 * A * cg15 * cg33 + 0.2D1 * t506 *
cg53
    # + 0.2D1 * t349 * cg18 + 0.2D1 * t352 * cg18 + 0.2D1 * t150 *
cg38
    # + 0.8D1 * t1133 * t482 + 0.24D2 * t74 * t1584 + 0.8D1 * t383 *
t1
    #446 + 0.8D1 * t383 * t1552 + 0.24D2 * t74 * t1588 + 0.24D2 *
t1105
    # * t513 * tlrho + 0.12D2 * t392 * cg53 * trho + 0.12D2 * t392 *
cg
    #33 * cg15 + 0.8D1 * t383 * t1578 + 0.12D2 * t392 * cg18 * tlrho +
#0.4D1 * t164 * cg38
        t1636 = t493 * tlrho
        t1648 = -0.400350D0 * t477 * t470 * t1564 + 0.266900D0 * t329 *
#t500 * t1497 + 0.133450D0 * t464 * t1571 - 0.133450D0 * t1053 *
t4
    #71 - 0.133450D0 * t1053 * t494 + 0.133450D0 * t197 * t1578 * t80
-

```

```

# 0.66725D-1 * t73 * t159 * t1627 - 0.66725D-1 * t73 * t992 * t206
#+ 0.133450D0 * t1049 * t501 - 0.133450D0 * t1004 * t1636 -
0.13345
#0D0 * t1068 * t471 + 0.133450D0 * t318 * t467 + 0.266900D0 * t329
#* t500 * t1564 - 0.133450D0 * t329 * t1517
t1683 = 0.133450D0 * t318 * t194 + 0.133450D0 * t145 * t1571 -
0
#.133450D0 * t329 * t1439 + 0.133450D0 * t145 * t1501 + 0.400350D0
#* t477 * t1484 + 0.133450D0 * t197 * t1425 * t80 + 0.133450D0 *
t1
#97 * A * cg53 * t80 - 0.133450D0 * t490 * t1439 - 0.133450D0 *
t32
#9 * t1636 - 0.66725D-1 * t73 * t369 * t206 + 0.133450D0 * t373 *
t
#1480 - 0.66725D-1 * t73 * t159 * t1433
t1684 = cg59 * t1683
t1686 = t1240 * t171
cg31 = cg47
t1702 = t536 * slrho
t1720 = t91 * cg47
cg26 = -0.9950248765D1 * t437 * t532 * slrho + 0.2000000001D1 *
#t186 * slrho * cg13 + 0.2000000001D1 * t186 * s * cg31
exc_norm_drho_rho_rho = cg24 * cg58 + cg16 * cg26 + cg12 * cg58
#+ t453 * cg58 + t190 * cg26 + cg16 * cg48 + t457 * cg48 + t97 *
(0
#.7425558783D2 * t1329 * t1702 - 0.2985074630D2 * t1333 * t1702 -
0
#.9950248765D1 * t437 * t91 * cg31 * srho - 0.9950248765D1 * t437
*
# t532 * cg57 + 0.2000000001D1 * t186 * cg57 * cg13 +
0.2000000001D
#1 * t186 * srho * cg31 - 0.9950248765D1 * t437 * t1720 * slrho +
0
#.2000000001D1 * t186 * slrho * cg47 + 0.2000000001D1 * t186 * s *
#(t423 * t6 * t425 + t178 * t102 * kfrho - t528 * cg32 / 0.2D1 +
t8
#8 * t224)) + t1684 * t173 - t211 * t416 + cg19 + rho * (cg59 *
(t1
#477 + t1520 + t1563 + t1648) * t173 - t523 * t416 - t1684 * t525
+
# 0.2D1 * t211 * t1686 * t415 - t211 * t406 * t402)
t1747 = t470 * t518
t1759 = t375 * t559
t1766 = Arho * t545
t1786 = t1442 * t206
t1794 = -0.266900D0 * t490 * t1747 - 0.800700D0 * t477 * t470 *
#t206 * trho + 0.1334500D1 * t477 * t478 * cg18 - 0.266900D0 *

```

t100

#4 * t553 + 0.266900D0 * t329 * t1759 * trho + 0.266900D0 * t373 *
#t1533 * t206 - 0.66725D-1 * t73 * t159 * (0.2D1 * t1766 + 0.4D1 *
#t486 * cg33 + 0.24D2 * t74 * t1766 + 0.24D2 * t1105 * t545 * trho
#+ 0.24D2 * t392 * cg18 * cg33) - 0.266900D0 * t1455 * t553 -

0.133

#450D0 * t546 * t168 - 0.266900D0 * t490 * t1786 - 0.266900D0 *

t32

#9 * t1786 + 0.266900D0 * t66 * cg18 * t194
t1806 = t500 * cg33 * t167
t1811 = t545 * t80
t1815 = t375 * t566
t1829 = t545 * t158
t1833 = t1167 * t559
t1837 = 0.133450D0 * t73 * t153 * t375 * t559 - 0.133450D0 *

t32

#9 * t158 * t566 * trho - 0.266900D0 * t1438 * t553 + 0.533800D0 *
#t329 * t1806 - 0.266900D0 * t329 * t1747 + 0.133450D1 * t1506 *

t

#1811 * trho + 0.133450D0 * t373 * t1815 * t167 + 0.133450D0 *

t546

* t154 + 0.533800D0 * t490 * t1806 - 0.66725D-1 * t73 * t361 *

t5

#66 + 0.667250D0 * t73 * t1766 * t80 - 0.667250D0 * t477 * t1829 *
#t167 - 0.400350D0 * t373 * t1833 * t167

t1842 = t406 * t210

t1848 = t1328 * t576

t1852 = s * t576

exc_norm_drho_norm_drho_rho = cg16 * cg28 + t190 * cg28 + t97 *
#(0.7425558783D2 * t1327 * t1848 * srho - 0.2985074630D2 * t437 *

t

#1852 * srho - 0.1990049753D2 * t437 * t1720 * cg13 +

0.4000000002D

#1 * t186 * cg47 * cg13) + cg9 + rho * (cg59 * (t1794 + t1837) *

t1

#73 - t571 * t525 - 0.2D1 * t523 * t1842 + 0.2D1 * t574 * t1686)

cg63 = cg33

t1865 = A * cg63

t1871 = 0.2D1 * t1865 * cg33 + 0.12D2 * t392 * cg33 * cg63

t1876 = cg63 * t158 * t566

t1883 = 0.2D1 * t150 * cg63 + 0.4D1 * t164 * cg63

t1885 = t1458 * t206 * t1883

t1891 = t158 * t1871 * cg33

t1910 = t1759 * cg63

t1913 = t66 * cg63

t1924 = t159 * t1883

t1932 = 0.266900D0 * t373 * t500 * t1871 - 0.133450D0 * t490 *

```

t
    #1876 + 0.533800D0 * t490 * t1885 + 0.533800D0 * t329 * t1885 -
0.2
    #66900D0 * t329 * t1891 - 0.266900D0 * t490 * t1891 + 0.133450D0 *
    #t373 * t1815 * t1883 - 0.1601400D1 * t197 * t75 * t158 * t76 *
t54
    #5 * cg63 - 0.133450D0 * t329 * t1876 + 0.1601400D1 * t1506 *
t1811
    # * cg63 + 0.266900D0 * t329 * t1910 - 0.266900D0 * t1913 * t75 *
t
    #553 - 0.133450D1 * t477 * t470 * t206 * cg63 - 0.667250D0 * t477
    #* t1829 * t1883 - 0.133450D0 * t546 * t1924 + 0.266900D0 * t490 *
    #t1910 - 0.400350D0 * t373 * t1833 * t1883
    t1943 = 0.133450D0 * t145 * t81 * cg63 + 0.133450D0 * t197 *
t18
    #65 * t80 - 0.66725D-1 * t73 * t1924
    t1951 = t470 * t1883
    t1956 = t493 * cg63
    cg8 = cg13
    exc_norm_drho_norm_drho_norm_drho = t97 * (0.7425558783D2 *
t132
    #7 * t1848 * cg8 - 0.2985074630D2 * t437 * t1852 * cg8) + rho *
(cg
    #59 * t1932 * t173 - t571 * t406 * t1943 - 0.2D1 * cg59 *
(0.133450
    #D0 * t1913 * t194 + 0.667250D0 * t477 * t478 * cg63 - 0.133450D0
*
    # t329 * t1951 - 0.133450D0 * t490 * t1951 - 0.133450D0 * t329 *
t1
    #956 - 0.133450D0 * t490 * t1956 + 0.133450D0 * t373 * t500 *
t1883
    # - 0.66725D-1 * t73 * t159 * t1871) * t1842 + 0.2D1 * t574 *
t1240
    # * t1943)
    cg66 = exc_norm_drho_norm_drho_norm_drho
    return
end

```

LSD

```

> loc(eqs_ex_lda)intersect loc(eqs_pbec2);
                                {β}

> loc(eqs_ex_lda);
                                {s, ex_lda, ex_unif, Fx, kf, μ, κ, β}

```

```
> toRn:=convert(loc(eqs_ex_lda)minus {beta,kappa,mu},list);
```

```
toRn := [s, ex_lda, ex_unif, Fx, kf]
```

```
> eqs_ex_a:=subs(op(map(x->x=x | |_a,toRn)),rho=2*rhoa,norm_drho=2*norm_drhoa,eqs_ex_lda);
```

```
eqs_ex_b:=subs(op(map(x->x=x | |_b,toRn)),rho=2*rhob,norm_drho=2*norm_drhob,eqs_ex_lda);
```

$$eqs_ex_a := \left[\kappa = 0.804, \beta = 0.066725, \mu = \frac{1}{3} \beta \pi^2, kf_a = 3^{(1/3)} 2^{(1/3)} (\pi^2 rhoa)^{(1/3)}, \right.$$

$$ex_unif_a = -\frac{3 kf_a}{4 \pi}, s_a = \frac{norm_drhoa}{2 kf_a rhoa}, Fx_a = 1 + \kappa - \frac{\kappa}{1 + \frac{\mu s_a^2}{\kappa}},$$

$$ex_lda_a = 2 rhoa ex_unif_a Fx_a \left. \right]$$

$$eqs_ex_b := \left[\kappa = 0.804, \beta = 0.066725, \mu = \frac{1}{3} \beta \pi^2, kf_b = 3^{(1/3)} 2^{(1/3)} (\pi^2 rhob)^{(1/3)}, \right.$$

$$ex_unif_b = -\frac{3 kf_b}{4 \pi}, s_b = \frac{norm_drhob}{2 kf_b rhob}, Fx_b = 1 + \kappa - \frac{\kappa}{1 + \frac{\mu s_b^2}{\kappa}},$$

$$ex_lda_b = 2 rhob ex_unif_b Fx_b \left. \right]$$

```
> unk(eqs_ex_a);
```

```
unk(eqs_ex_b);
```

```
{pi, norm_drhoa, rhoa}
```

```
{pi, norm_drhob, rhob}
```

```
> sameNameSameDef(eqs_pbec2,eqs_ex_a);
```

```
sameNameSameDef(eqs_pbec2,eqs_ex_b);
```

```
sameNameSameDef(eqs_ex_a,eqs_ex_b);
```

```
true
```

```
true
```

```
true
```

```
> eqs_lsd:=combineEqs([rho=rhoa+rhob],eqs_pbec2,eqs_ex_a,eqs_ex_b,[exc=(ex_lda_a+ex_lda_b)/2+ec]):
```

```
unk(eqs_1sd);
```

```
{ $\pi$ , norm_drhoa, norm_drho, norm_drhob, rhoa, rhob}
```

```
> arg_1sd_names:=[rhoa,rhob,norm_drho,norm_drhoa,norm_drhob];  
arg_1sd_names := [rhoa, rhob, norm_drho, norm_drhoa, norm_drhob]
```

```
> eqs_1sd2:=sostConst(eqs_1sd):  
unk(eqs_1sd2);  
{ $\pi$ , norm_drhoa, norm_drho, norm_drhob, rhoa, rhob}
```

```
> deriv_1sd1:=calcDerivs(eqs_1sd2,arg_1sd_names):
```

```
> deriv_1sd2:=[seq(seq(op(calcDerivs(deriv_1sd1[i],[arg_1sd_names[j]])),i  
=1..j),j=1..nops(arg_1sd_names))]:
```

```
> #deriv_1sd3:=[seq(seq(op(calcDerivs(deriv_1sd2[i],[arg_1sd_names[j]])),  
i=1..((j+1)*j/2)),j=1..nops(arg_1sd_names))]:
```

```
> n_deriv1:=map(x->lhs(op(-1,x)),deriv_1sd1);  
n_deriv2:=map(x->lhs(op(-1,x)),deriv_1sd2);  
#n_deriv3:=map(x->lhs(op(-1,x)),deriv_1sd3);  
n_deriv1 := [exc_rhoa, exc_rhob, exc_norm_drho, exc_norm_drhoa, exc_norm_drhob]
```

```
n_deriv2 := [exc_rhoa_rhoa, exc_rhoa_rhob, exc_rhob_rhob, exc_rhoa_norm_drho,  
exc_rhob_norm_drho, exc_norm_drho_norm_drho, exc_rhoa_norm_drhoa, exc_rhob_norm_drhoc,  
exc_norm_drho_norm_drhoa, exc_norm_drhoa_norm_drhoa, exc_rhoa_norm_drhob,  
exc_rhob_norm_drhob, exc_norm_drho_norm_drhob, exc_norm_drhoa_norm_drhob,  
exc_norm_drhob_norm_drhob]
```

```
> 'sameNameSameDef(eqs_1sd2,deriv_1sd1[i])'$i=1..nops(deriv_1sd1);  
'sameNameSameDef(eqs_1sd2,deriv_1sd2[i])'$i=1..nops(deriv_1sd1);  
#foldl(`and`,sameNameSameDef(eqs_1sd2,deriv_1sd3[i])$i=1..nops(deriv_ls  
d3));  
foldl(`and`,seq(sameNameSameDef(deriv_1sd1[i],deriv_1sd2[j])$i=1..nops(  
deriv_1sd1),j=1..nops(deriv_1sd2)));  
#foldl(`and`,seq(sameNameSameDef(deriv_1sd1[i],deriv_1sd3[j])$i=1..nops  
(deriv_1sd1),j=1..nops(deriv_1sd3)));  
#foldl(`and`,seq(sameNameSameDef(deriv_1sd2[i],deriv_1sd3[j])$i=1..nops  
(deriv_1sd2),j=1..nops(deriv_1sd3)));
```

```
true, true, true, true, true
```

```
true, true, true, true, true
```

```
true
```

```
> eqs_1sd3:=combineEqs([eqs_1sd2,op(deriv_1sd1),op(deriv_1sd2)]): #  
,op(deriv_1sd3) # removed 3 deriv
```

```
> eqs_1sd4:=enforceDependencies([my_rhoa=rhoa,my_rhob=rhob,  
  
my_norm_drho=norm_drho,my_norm_drhoa=norm_drhoa,my_norm_drhob=norm_drho  
b,  
op(subs(rhoa=my_rhoa,rhob=my_rhob,norm_drho=my_norm_drho,
```

```

        norm_drhoa=my_norm_drhoa,norm_drhob=my_norm_drhob,eqs_1sd3))):
> res_eqs_1sd:=[exc,op(n_deriv1),op(n_deriv2)]:#,op(n_deriv3)]: # removed
3 deriv
res_eqs_1sd2:=[]:
for my_symb in res_eqs_1sd do
    if not rhs(getDef(my_symb,eqs_1sd4))=0 then
        res_eqs_1sd2:=[op(res_eqs_1sd2),my_symb]:
    end if;
end do;

> for my_symb in res_eqs_1sd2 do
    print(my_symb,unk([op(eqs_1sd4),result=my_symb]));
end do;
my_symb:='my_symb':
    exc, { $\pi$ , norm_drhoa, norm_drho, norm_drhob, rhoa, rhob}
    exc_rhoa, { $\pi$ , norm_drhoa, norm_drho, rhoa, rhob}
    exc_rhob, { $\pi$ , norm_drho, norm_drhob, rhoa, rhob}
    exc_norm_drho, { $\pi$ , norm_drho, rhoa, rhob}
    exc_norm_drhoa, { $\pi$ , norm_drhoa, rhoa}
    exc_norm_drhob, { $\pi$ , norm_drhob, rhob}
    exc_rhoa_rhoa, { $\pi$ , norm_drhoa, norm_drho, rhoa, rhob}
    exc_rhoa_rhob, { $\pi$ , norm_drho, rhoa, rhob}
    exc_rhob_rhob, { $\pi$ , norm_drho, norm_drhob, rhoa, rhob}
    exc_rhoa_norm_drho, { $\pi$ , norm_drho, rhoa, rhob}
    exc_rhob_norm_drho, { $\pi$ , norm_drho, rhoa, rhob}
    exc_norm_drho_norm_drho, { $\pi$ , norm_drho, rhoa, rhob}
    exc_rhoa_norm_drhoa, { $\pi$ , norm_drhoa, rhoa}
    exc_norm_drhoa_norm_drhoa, { $\pi$ , norm_drhoa, rhoa}
    exc_rhob_norm_drhob, { $\pi$ , norm_drhob, rhob}
    exc_norm_drhob_norm_drhob, { $\pi$ , norm_drhob, rhob}

> for i from 1 to nops(eqs_1sd3) do
    for j from i+1 to nops(eqs_1sd3) do
        if eqUses(eqs_1sd3[i],eqs_1sd3[j]) then
            print(i,j,eqs_1sd3[i],uses,eqs_1sd3[j]);
        end if;
    end do;
end do;
i:='i':j:='j':

> glob_eqs_1sd4:=[my_rhoa,my_rhob,my_norm_drho,my_norm_drhoa,my_norm_drho
b,op(res_eqs_1sd2)];
args_1sd:=[rhoa,rhob,norm_drho,norm_drhoa,norm_drhob];

```



```
glob_eqs_lsd4 := [my_rhoa, my_rhob, my_norm_drho, my_norm_drhoa, my_norm_drhob, exc, exc_rhoa,
exc_rhob, exc_norm_drho, exc_norm_drhoa, exc_norm_drhob, exc_rhoa_rhoa, exc_rhoa_rhob,
exc_rhob_rhob, exc_rhoa_norm_drho, exc_rhob_norm_drho, exc_norm_drho_norm_drho,
exc_rhoa_norm_drhoa, exc_norm_drhoa_norm_drhoa, exc_rhob_norm_drhob,
exc_norm_drhob_norm_drhob]
```

```
args_lsd := [rhoa, rhob, norm_drho, norm_drhoa, norm_drhob]
```

```
> cs_eqs_lsd4:=CompSeq(locals=loc2(eqs_lsd4)minus
convert(glob_eqs_lsd4,set),
globals=convert(glob_eqs_lsd4,list),params=args_lsd,eqs_lsd4):
r_eqs_lsd4:=convert(cs_eqs_lsd4,procedure):
```

Fortran code

```
> Fortran(r_eqs_lsd4,defaulttype=float,optimize);
```

```
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", "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"] = ["norm_drho", "norm_drhoa",
"norm_drhob", "chirhobrhob", "epsilon_c_unif", "s_anorm_drhoa",
"epsilon_c_unifrhoa", "alpha_clrhob", "alpha_clrhoa",
"epsilon_c_uniflrhoa", "alpha_crhob", "epsilon_c_unifrhob", "alpha_c",
"kf_brhobrhob", "epsilon_cGGArhoa", "rsrhobrhob", "epsilon_cGGArhob",
"Fx_brhob", "Fx_a", "phirhobrhob", "k_slrhob", "gamma_var",
"ex_unif_brhob", "kf_b", "s_a", "k_slrhoa", "tnorm_drho", "trhoarhob",
"Fx_b", "kf_arhoarhoa", "ex_unif_alrhoa", "kf_brhob", "Fx_alrhoa",
"chirhoa", "phirhoarhob", "k_srhoa", "chirhoarhoa", "epsilon_cGGA",
"s_b", "phirlhob", "Fx_blrhob", "s_blrhob", "s_brhob", "e_c_u_0",
"e_c_u_0rhoa", "epsilon_c_unifrhoarhoa", "chirhoarhob",
"e_c_u_0rhobrhob", "trhoanorm_drho", "frhoarhob", "frhoarhoa",
"alpha_crhoa", "Arhobrhob", "rsrhoarhob", "phirhoa",
"epsilon_c_unifrhobrhob", "phirhob", "ex_unif_arhoa", "k_frhob",
"frhobrhob", "kf_arhoa", "k_s", "epsilon_c_unifrhoarhob",
"e_c_u_lrhob", "k_frhoa", "k_srhob", "ex_unif_a", "rsrhoarhoa",
"e_c_u_0lrhoa", "phirlrhoa", "e_c_u_0lrhob", "s_alrhoa", "Fx_arhoa",
"e_c_u_lrhoa", "s_arhoa", "chirhob", "k_frhoarhob", "trhoarhoa",
"phirhoarhoa", "ex_unif_b", "ex_unif_blrhob", "e_c_u_0rhoarhob",
"Arhoarhob", "Arhoarhoa", "Hnorm_drho", "epsilon_c_uniflrhob",
"trhobnorm_drho", "Fx_anorm_drhoa", "s_bnorm_drhob", "Fx_bnorm_drhob",
"trhobrhob", "e_c_u_0rhoarhoa", "kf_a", "k_frhoarhoa", "e_c_u_0rhob",
"r eqs lsd4"]
```

```

doubleprecision function cg94 (rhoa, rhob, cg, cg0, cg1)
  doubleprecision my_rhoa
  doubleprecision my_rhob
  doubleprecision my_norm_drho
  doubleprecision my_norm_drhoa
  doubleprecision my_norm_drhob
  doubleprecision exc
  doubleprecision exc_rhoa
  doubleprecision exc_rhob
  doubleprecision exc_norm_drho
  doubleprecision exc_norm_drhoa
  doubleprecision exc_norm_drhob
  doubleprecision exc_rhoa_rhoa
  doubleprecision exc_rhoa_rhob
  doubleprecision exc_rhob_rhob
  doubleprecision exc_rhoa_norm_drho
  doubleprecision exc_rhob_norm_drho
  doubleprecision exc_norm_drho_norm_drho
  doubleprecision exc_rhoa_norm_drhoa
  doubleprecision exc_norm_drhoa_norm_drhoa
  doubleprecision exc_rhob_norm_drhob
  doubleprecision exc_norm_drhob_norm_drhob
  common my_rhoa, my_rhob, my_norm_drho, my_norm_drhoa,
my_norm_dr
    #hob, exc, exc_rhoa, exc_rhob, exc_norm_drho, exc_norm_drhoa,
exc_n
    #orm_drhob, exc_rhoa_rhoa, exc_rhoa_rhob, exc_rhob_rhob,
exc_rhoa_n
    #orm_drho, exc_rhob_norm_drho, exc_norm_drho_norm_drho,
exc_rhoa_no
    #rm_drhoa, exc_norm_drhoa_norm_drhoa, exc_rhob_norm_drhob,
exc_norm
    #_drhob_norm_drhob
    doubleprecision rhoa
    doubleprecision rhob
    doubleprecision cg
    doubleprecision cg0
    doubleprecision cg1
    doubleprecision t119
    doubleprecision t290
    doubleprecision t105
    doubleprecision t1514
    doubleprecision t107
    doubleprecision t1552
    doubleprecision t104
    doubleprecision cg2

```

	doubleprecision	t1
	doubleprecision	cg3
	doubleprecision	t1717
	doubleprecision	t118
	doubleprecision	t724
	doubleprecision	cg4
	doubleprecision	t102
	doubleprecision	t103
	doubleprecision	t898
	doubleprecision	t122
	doubleprecision	t538
	doubleprecision	t897
	doubleprecision	t2
	doubleprecision	t101
	doubleprecision	t80
	doubleprecision	t304
	doubleprecision	t181
	doubleprecision	t776
	doubleprecision	t125
	doubleprecision	t100
	doubleprecision	t70
	doubleprecision	t128
	doubleprecision	t366
	doubleprecision	t368
	doubleprecision	cg5
	doubleprecision	t20
	doubleprecision	rsrhoa
	doubleprecision	t81
	doubleprecision	t507
	doubleprecision	t510
	doubleprecision	t864
	doubleprecision	t130
	doubleprecision	t1277
	doubleprecision	cg6
	doubleprecision	t50
	doubleprecision	t772
	doubleprecision	t120
	doubleprecision	t632
	doubleprecision	t167
	doubleprecision	cg7
	integer	t3
	doubleprecision	t134
	doubleprecision	t249
	doubleprecision	t250
	doubleprecision	t251
	doubleprecision	t133
	doubleprecision	t427

	doubleprecision	t412
	doubleprecision	t1214
	doubleprecision	t204
	doubleprecision	trhoa
	doubleprecision	t196
	doubleprecision	t241
	doubleprecision	t242
	doubleprecision	t135
	doubleprecision	t223
	doubleprecision	t1426
	doubleprecision	t1428
	doubleprecision	t1429
	doubleprecision	t573
	doubleprecision	cg8
	doubleprecision	t71
	doubleprecision	t137
	doubleprecision	t429
	doubleprecision	t354
	doubleprecision	t82
	doubleprecision	cg9
	doubleprecision	cg10
	doubleprecision	t293
	doubleprecision	t378
	doubleprecision	t41
	doubleprecision	t83
	doubleprecision	t685
	doubleprecision	t387
	doubleprecision	cg11
	doubleprecision	cg12
	doubleprecision	t659
	doubleprecision	t663
	doubleprecision	t664
	doubleprecision	t168
	doubleprecision	t169
	doubleprecision	t12
	doubleprecision	t1032
	doubleprecision	t143
	doubleprecision	t305
	doubleprecision	t308
	doubleprecision	t813
	doubleprecision	t37
	doubleprecision	t937
	doubleprecision	cg13
	doubleprecision	t274
	doubleprecision	cg14
	doubleprecision	t1194
	doubleprecision	cg15

	doubleprecision	cg16
	doubleprecision	cg17
	doubleprecision	t95
	doubleprecision	cg18
	doubleprecision	cg19
	doubleprecision	t824
	doubleprecision	cg20
	doubleprecision	t577
	doubleprecision	t517
	doubleprecision	cg21
	doubleprecision	t73
	doubleprecision	t275
	doubleprecision	t1476
	doubleprecision	t1478
	integer	t4
	doubleprecision	cg22
	doubleprecision	t1575
	doubleprecision	t115
	doubleprecision	t1219
	doubleprecision	t1731
	doubleprecision	t1221
	doubleprecision	t1224
	doubleprecision	t1437
	doubleprecision	t1441
	doubleprecision	t1442
	doubleprecision	t1061
	doubleprecision	t65
	doubleprecision	t857
	doubleprecision	t865
	doubleprecision	t866
	doubleprecision	t13
	doubleprecision	t96
	doubleprecision	t380
	doubleprecision	t383
	doubleprecision	t346
	doubleprecision	t347
	doubleprecision	t144
	doubleprecision	t140
	doubleprecision	phi
	integer	t55
	doubleprecision	t1209
	doubleprecision	t1213
	doubleprecision	t145
	doubleprecision	t77
	doubleprecision	t114
	doubleprecision	t1237
	doubleprecision	Alrhoa

	doubleprecision	cg23
	doubleprecision	t934
	doubleprecision	t939
	doubleprecision	t941
	doubleprecision	t943
	doubleprecision	cg24
	doubleprecision	t675
	doubleprecision	t681
	doubleprecision	cg25
	doubleprecision	t1577
	doubleprecision	t276
	doubleprecision	t277
	doubleprecision	t735
	doubleprecision	t736
	doubleprecision	t601
	doubleprecision	t604
	doubleprecision	t777
	doubleprecision	t420
	doubleprecision	t423
	doubleprecision	t336
	doubleprecision	t337
	doubleprecision	cg26
	doubleprecision	t982
	doubleprecision	t1706
	doubleprecision	t146
	doubleprecision	t569
	doubleprecision	t571
	doubleprecision	t572
	doubleprecision	t842
	doubleprecision	t1357
	doubleprecision	t299
	doubleprecision	t300
	doubleprecision	t301
	integer	t5
	doubleprecision	t1198
	doubleprecision	t1553
	doubleprecision	t23
	doubleprecision	t1579
	doubleprecision	cg27
	doubleprecision	t1539
	doubleprecision	t91
	doubleprecision	t1319
	doubleprecision	t1326
	doubleprecision	t159
	doubleprecision	t553
	doubleprecision	t554
	doubleprecision	cg28

	doubleprecision A
	doubleprecision t66
	doubleprecision chi
	doubleprecision Arhoa
	integer t6
	doubleprecision t404
	doubleprecision t408
	doubleprecision t409
	doubleprecision t450
	doubleprecision t455
	doubleprecision t33
	doubleprecision cg29
	doubleprecision t390
	doubleprecision t1299
	doubleprecision t218
	doubleprecision t220
	doubleprecision flrhob
	doubleprecision cg30
	doubleprecision t578
	doubleprecision t1696
	doubleprecision t24
	doubleprecision t418
	doubleprecision t419
	doubleprecision t314
	doubleprecision cg31
	doubleprecision t84
	doubleprecision t85
	doubleprecision t834
	doubleprecision t957
	doubleprecision t319
	doubleprecision cg32
	doubleprecision t717
	doubleprecision t622
	doubleprecision flrhoa
	doubleprecision t1615
	doubleprecision t1158
	doubleprecision t1160
	doubleprecision cg33
	doubleprecision t925
	doubleprecision cg34
	doubleprecision cg35
	doubleprecision t188
	doubleprecision t189
	doubleprecision t190
	doubleprecision t263
	doubleprecision cg36
	doubleprecision cg37

	doubleprecision	t247
	doubleprecision	t815
	doubleprecision	t820
	doubleprecision	t7
	doubleprecision	Arhob
	doubleprecision	t861
	doubleprecision	t463
	doubleprecision	t464
	doubleprecision	cg38
	doubleprecision	t509
	doubleprecision	t607
	doubleprecision	t1584
	doubleprecision	t63
	doubleprecision	Alrhob
	doubleprecision	t1150
	doubleprecision	t1153
	doubleprecision	t1155
	doubleprecision	t254
	doubleprecision	t86
	doubleprecision	t232
	doubleprecision	t233
	doubleprecision	t629
	doubleprecision	t899
	doubleprecision	t807
	doubleprecision	t289
	doubleprecision	t454
	doubleprecision	t671
	doubleprecision	t672
	doubleprecision	t832
	doubleprecision	cg39
	doubleprecision	t1702
	doubleprecision	t424
	doubleprecision	t212
	doubleprecision	t540
	doubleprecision	cg40
	doubleprecision	t147
	doubleprecision	cg41
	doubleprecision	t433
	doubleprecision	t92
	doubleprecision	t149
	doubleprecision	t362
	doubleprecision	t364
	doubleprecision	t244
	doubleprecision	t246
	doubleprecision	t851
	doubleprecision	t708
	doubleprecision	t56

	doubleprecision	t87
	doubleprecision	t794
	doubleprecision	cg42
	doubleprecision	t754
	doubleprecision	cg43
	doubleprecision	t1587
	doubleprecision	t68
	doubleprecision	t511
	doubleprecision	t442
	doubleprecision	t1468
	doubleprecision	t1470
	doubleprecision	cg44
	doubleprecision	t1588
	doubleprecision	cg45
	doubleprecision	t310
	doubleprecision	mu
	doubleprecision	t1458
	doubleprecision	t258
	doubleprecision	cg46
	doubleprecision	t1096
	doubleprecision	t1099
	doubleprecision	t360
	doubleprecision	t1259
	doubleprecision	t479
	doubleprecision	t57
	doubleprecision	t932
	doubleprecision	cg47
	doubleprecision	t99
	doubleprecision	t58
	doubleprecision	rs
	doubleprecision	cg48
	doubleprecision	cg49
	doubleprecision	t356
	doubleprecision	t358
	doubleprecision	t489
	doubleprecision	t490
	doubleprecision	t491
	doubleprecision	t493
	doubleprecision	t496
	doubleprecision	cg50
	doubleprecision	cg51
	doubleprecision	t1636
	doubleprecision	t8
	doubleprecision	t1338
	doubleprecision	cg52
	doubleprecision	frhoa
	doubleprecision	cg53

	doubleprecision	t1750
	doubleprecision	t208
	doubleprecision	t209
	doubleprecision	t210
	doubleprecision	t59
	doubleprecision	t395
	doubleprecision	t441
	doubleprecision	t112
	doubleprecision	t361
	doubleprecision	cg54
	doubleprecision	cg55
	doubleprecision	t484
	doubleprecision	t129
	doubleprecision	t113
	doubleprecision	t262
	doubleprecision	t108
	doubleprecision	t322
	doubleprecision	t324
	doubleprecision	t326
	doubleprecision	t327
	doubleprecision	t9
	doubleprecision	t89
	doubleprecision	t1152
	doubleprecision	t1216
	doubleprecision	t284
	doubleprecision	t221
	doubleprecision	t719
	doubleprecision	t721
	doubleprecision	cg56
	doubleprecision	t485
	doubleprecision	cg57
	doubleprecision	t1662
	doubleprecision	cg58
	doubleprecision	t46
	doubleprecision	t761
	doubleprecision	t763
	doubleprecision	t764
	doubleprecision	t765
	doubleprecision	t75
	doubleprecision	cg59
	doubleprecision	cg60
	doubleprecision	trhob
	doubleprecision	t60
	doubleprecision	t1772
	doubleprecision	cg61
	doubleprecision	t716
	doubleprecision	t718

	doubleprecision	f
	doubleprecision	t646
	doubleprecision	t651
	doubleprecision	t1366
	doubleprecision	t61
	doubleprecision	cg62
	doubleprecision	t1663
	doubleprecision	cg63
	doubleprecision	cg64
	doubleprecision	t469
	doubleprecision	frhob
	doubleprecision	t155
	doubleprecision	t36
	doubleprecision	cg65
	doubleprecision	cg66
	doubleprecision	t436
	doubleprecision	t156
	doubleprecision	t1494
	doubleprecision	t1673
	doubleprecision	t78
	doubleprecision	t793
	doubleprecision	t1473
	doubleprecision	t541
	doubleprecision	t548
	doubleprecision	cg67
	doubleprecision	t157
	doubleprecision	t16
	doubleprecision	cg68
	integer	t52
	doubleprecision	cg69
	doubleprecision	cg70
	doubleprecision	cg71
	doubleprecision	t498
	doubleprecision	t1486
	doubleprecision	t1664
	doubleprecision	cg72
	doubleprecision	t236
	doubleprecision	t237
	doubleprecision	t238
	doubleprecision	cg73
	doubleprecision	t109
	doubleprecision	cg74
	doubleprecision	t206
	doubleprecision	t1329
	doubleprecision	t162
	doubleprecision	t164
	doubleprecision	cg75

	doubleprecision	t165
	doubleprecision	t213
	doubleprecision	t214
	doubleprecision	t216
	doubleprecision	t1430
	doubleprecision	cg76
	doubleprecision	t1rhoa
	doubleprecision	t195
	doubleprecision	t197
	doubleprecision	t
	doubleprecision	t18
	doubleprecision	cg77
	doubleprecision	t1351
	doubleprecision	t1518
	doubleprecision	t396
	doubleprecision	t76
	doubleprecision	t64
	doubleprecision	t280
	doubleprecision	t281
	doubleprecision	t288
	doubleprecision	t522
	doubleprecision	t523
	doubleprecision	cg78
	doubleprecision	t542
	doubleprecision	t1387
	doubleprecision	t1389
	doubleprecision	t1392
	doubleprecision	rho
	doubleprecision	cg79
	doubleprecision	cg80
	doubleprecision	cg81
	doubleprecision	t28
	doubleprecision	t266
	doubleprecision	t267
	doubleprecision	t269
	doubleprecision	t270
	doubleprecision	t271
	doubleprecision	cg82
	doubleprecision	cg83
	doubleprecision	t226
	doubleprecision	t230
	doubleprecision	rsrhob
	doubleprecision	t90
	doubleprecision	t501
	doubleprecision	cg84
	doubleprecision	t393
	doubleprecision	t397

```
doubleprecision t804
doubleprecision t805
doubleprecision t810
doubleprecision t259
doubleprecision cg85
doubleprecision t49
doubleprecision t1171
doubleprecision cg86
doubleprecision t158
doubleprecision t1612
doubleprecision t641
doubleprecision t884
doubleprecision t1601
doubleprecision cg87
doubleprecision cg88
doubleprecision cg89
doubleprecision cg90
doubleprecision t1521
doubleprecision t835
doubleprecision cg91
doubleprecision t1602
doubleprecision t174
doubleprecision t175
doubleprecision t176
doubleprecision t182
doubleprecision t183
doubleprecision t1603
doubleprecision t1302
doubleprecision t1711
doubleprecision t1rhob
doubleprecision cg92
doubleprecision t691
doubleprecision t694
doubleprecision t697
doubleprecision t93
doubleprecision cg93
doubleprecision t583
my_rhoa = rhoa
my_rhob = rhob
my_norm_drho = cg
my_norm_drhoa = cg0
my_norm_drhob = cg1
rho = rhoa + rhob
t1 = rhoa - rhob
t2 = 0.1D1 / rho
chi = t1 * t2
t3 = 3 ** (0.1D1 / 0.3D1)
```

```

t4 = 4 ** (0.1D1 / 0.3D1)
t5 = t4 ** 2
t6 = t3 * t5
t7 = 0.1D1 / 0.3141592654D1
t8 = t7 * t2
t9 = t8 ** (0.1D1 / 0.3D1)
rs = dble(t6) * t9 / 0.4D1
t12 = 0.1D1 + 0.21370D0 * rs
t13 = sqrt(rs)
t16 = t13 * rs
t18 = rs ** 0.20D1
t20 = 0.75957D1 * t13 + 0.35876D1 * rs + 0.16382D1 * t16 +
0.492
#94D0 * t18
t23 = 0.1D1 + 0.1608182432D2 / t20
t24 = log(t23)
cg47 = -0.62182D-1 * t12 * t24
t28 = 0.1D1 + 0.20548D0 * rs
t33 = 0.141189D2 * t13 + 0.61977D1 * rs + 0.33662D1 * t16 +
0.62
#517D0 * t18
t36 = 0.1D1 + 0.3216468318D2 / t33
t37 = log(t36)
t41 = 0.1D1 + 0.11125D0 * rs
t46 = 0.10357D2 * t13 + 0.36231D1 * rs + 0.88026D0 * t16 +
0.496
#71D0 * t18
t49 = 0.1D1 + 0.2960857464D1 / t46
t50 = log(t49)
cg19 = 0.33774D0 * t41 * t50
t52 = 2 ** (0.1D1 / 0.3D1)
t55 = 1 / (2 * t52 - 2)
t56 = 0.1D1 + chi
t57 = t56 ** (0.1D1 / 0.3D1)
t58 = t57 * t56
t59 = 0.1D1 - chi
t60 = t59 ** (0.1D1 / 0.3D1)
t61 = t60 * t59
f = (t58 + t61 - 0.2D1) * dble(t55)
t63 = cg19 * f
t64 = 0.9D1 / 0.8D1 / dble(t55)
t65 = chi ** 2
t66 = t65 ** 2
t68 = t64 * (0.1D1 - t66)
t70 = -0.31090D-1 * t28 * t37 - cg47
t71 = t70 * f
cg11 = cg47 + t63 * t68 + t71 * t66

```

```

t73 = log(0.2D1)
t75 = 0.3141592654D1 ** 2
t76 = 0.1D1 / t75
cg27 = (0.1D1 - t73) * t76
t77 = t57 ** 2
t78 = t60 ** 2
phi = t77 / 0.2D1 + t78 / 0.2D1
t80 = t75 * rho
t81 = t80 ** (0.1D1 / 0.3D1)
t82 = dble(t3) * t81 * t7
t83 = sqrt(t82)
cg63 = 0.2D1 * t83
t84 = 0.1D1 / phi
t85 = cg * t84
t86 = 0.1D1 / cg63
t87 = t86 * t2
t = t85 * t87 / 0.2D1
t89 = 0.1D1 / cg27
t90 = cg11 * t89
t91 = phi ** 2
t92 = t91 * phi
t93 = 0.1D1 / t92
t95 = exp(-t90 * t93)
t96 = t95 - 0.1D1
A = 0.66725D-1 * t89 / t96
t99 = cg27 * t92
t100 = t ** 2
t101 = t89 * t100
t102 = A * t100
t103 = 0.1D1 + t102
t104 = A ** 2
t105 = t100 ** 2
t107 = 0.1D1 + t102 + t104 * t105
t108 = 0.1D1 / t107
t109 = t103 * t108
t112 = 0.1D1 + 0.66725D-1 * t101 * t109
t113 = log(t112)
cg41 = cg11 + t99 * t113
mu = 0.2224166667D-1 * t75
t114 = t75 * rhoa
t115 = t114 ** (0.1D1 / 0.3D1)
cg91 = dble(t3) * t115
cg68 = -0.3D1 / 0.4D1 * t7 * cg91
t118 = 0.1D1 / cg91
t119 = cg0 * t118
t120 = 0.1D1 / rhoa
cg3 = t119 * t120 / 0.2D1

```

```

t122 = cg3 ** 2
t125 = 0.1D1 + 0.1243781095D1 * mu * t122
cg24 = 0.1804D1 - 0.804D0 / t125
t128 = rhoa * cg68
t129 = t75 * rhob
t130 = t129 ** (0.1D1 / 0.3D1)
cg29 = dble(t3) * t130
cg8 = -0.3D1 / 0.4D1 * t7 * cg29
t133 = 0.1D1 / cg29
t134 = cg1 * t133
t135 = 0.1D1 / rhob
cg42 = t134 * t135 / 0.2D1
t137 = cg42 ** 2
t140 = 0.1D1 + 0.1243781095D1 * mu * t137
cg33 = 0.1804D1 - 0.804D0 / t140
t143 = rhob * cg8
exc = t128 * cg24 + t143 * cg33 + rho * cg41
t144 = rho ** 2
t145 = 0.1D1 / t144
t146 = t1 * t145
cg38 = t2 - t146
t147 = t9 ** 2
t149 = 0.1D1 / t147 * t7
rsrhoa = -dble(t6) * t149 * t145 / 0.12D2
t155 = t20 ** 2
t156 = 0.1D1 / t155
t157 = t12 * t156
t158 = 0.1D1 / t13
t159 = t158 * rsrhoa
t162 = t13 * rsrhoa
t164 = rs ** 0.10D1
t165 = t164 * rsrhoa
t167 = 0.3797850000D1 * t159 + 0.35876D1 * rsrhoa +
0.2457300000
#D1 * t162 + 0.985880D0 * t165
t168 = 0.1D1 / t23
t169 = t167 * t168
cg48 = -0.1328829340D-1 * rsrhoa * t24 + 0.9999999999D0 * t157
*
# t169
t174 = t33 ** 2
t175 = 0.1D1 / t174
t176 = t28 * t175
t181 = 0.7059450000D1 * t159 + 0.61977D1 * rsrhoa +
0.5049300000
#D1 * t162 + 0.1250340D1 * t165
t182 = 0.1D1 / t36

```



```

t183 = t181 * t182
cg74 = -0.638837320D-2 * rsrhoa * t37 + 0.1000000000D1 * t176 *
#t183
t188 = t46 ** 2
t189 = 0.1D1 / t188
t190 = t41 * t189
t195 = 0.5178500000D1 * t159 + 0.36231D1 * rsrhoa +
0.1320390000
#D1 * t162 + 0.993420D0 * t165
t196 = 0.1D1 / t49
t197 = t195 * t196
cg54 = 0.375735750D-1 * rsrhoa * t50 - 0.9999999999D0 * t190 *
t
#197
frhoa = (0.4D1 / 0.3D1 * t57 * cg38 - 0.4D1 / 0.3D1 * t60 *
cg38
#) * dble(t55)
t204 = cg54 * f
t206 = cg19 * frhoa
t208 = t65 * chi
t209 = t64 * t208
t210 = t209 * cg38
t212 = 0.4D1 * t63 * t210
t213 = cg74 - cg48
t214 = t213 * f
t216 = t70 * frhoa
t218 = t208 * cg38
t220 = 0.4D1 * t71 * t218
cg13 = cg48 + t204 * t68 + t206 * t68 - t212 + t214 * t66 +
t216
# * t66 + t220
t221 = 0.1D1 / t57
t223 = 0.1D1 / t60
cg57 = t221 * cg38 / 0.3D1 - t223 * cg38 / 0.3D1
t226 = t81 ** 2
cg66 = dble(t3) / t226 * t75 / 0.3D1
t230 = 0.1D1 / t83
cg4 = t230 * cg66 * t7
t232 = 0.1D1 / t91
t233 = cg * t232
t236 = cg63 ** 2
t237 = 0.1D1 / t236
t238 = t237 * t2
t241 = t86 * t145
t242 = t85 * t241
trhoa = -t233 * t87 * cg57 / 0.2D1 - t85 * t238 * cg4 / 0.2D1 -
#t242 / 0.2D1

```

```

t244 = t96 ** 2
t246 = t89 / t244
t247 = cg13 * t89
t249 = t91 ** 2
t250 = 0.1D1 / t249
t251 = t250 * cg57
t254 = -t247 * t93 + 0.3D1 * t90 * t251
Arhoa = -0.66725D-1 * t246 * t254 * t95
t258 = cg27 * t91
t259 = t113 * cg57
t262 = t89 * t
t263 = t109 * trhoa
t266 = Arhoa * t100
t267 = A * t
t269 = 0.2D1 * t267 * trhoa
t270 = t266 + t269
t271 = t270 * t108
t274 = t107 ** 2
t275 = 0.1D1 / t274
t276 = t103 * t275
t277 = A * t105
t280 = t100 * t
t281 = t104 * t280
t284 = t266 + t269 + 0.2D1 * t277 * Arhoa + 0.4D1 * t281 *
trhoa
t288 = 0.133450D0 * t262 * t263 + 0.66725D-1 * t101 * t271 -
0.6
#6725D-1 * t101 * t276 * t284
t289 = 0.1D1 / t112
t290 = t288 * t289
cg20 = cg13 + 0.3D1 * t258 * t259 + t99 * t290
t293 = t115 ** 2
cg62 = dble(t3) / t293 * t75 / 0.3D1
cg6 = -0.3D1 / 0.4D1 * t7 * cg62
t299 = cg91 ** 2
t300 = 0.1D1 / t299
t301 = cg0 * t300
t304 = rhoa ** 2
t305 = 0.1D1 / t304
cg75 = -t301 * t120 * cg62 / 0.2D1 - t119 * t305 / 0.2D1
t308 = t125 ** 2
t310 = 0.1D1 / t308 * mu
cg73 = 0.2000000001D1 * t310 * cg3 * cg75
t314 = rhoa * cg6
exc_rhoa = cg68 * cg24 + t314 * cg24 + t128 * cg73 + cg41 + rho
#* cg20
cg76 = -t2 - t146

```

```

        rsrhob = rsrhoa
        t319 = t158 * rsrhob
        t322 = t13 * rsrhob
        t324 = t164 * rsrhob
        t326 = 0.3797850000D1 * t319 + 0.35876D1 * rsrhob +
0.2457300000
        #D1 * t322 + 0.985880D0 * t324
        t327 = t326 * t168
        cg93 = -0.1328829340D-1 * rsrhob * t24 + 0.9999999999D0 * t157
*
        # t327
        t336 = 0.7059450000D1 * t319 + 0.61977D1 * rsrhob +
0.5049300000
        #D1 * t322 + 0.1250340D1 * t324
        t337 = t336 * t182
        cg65 = -0.638837320D-2 * rsrhob * t37 + 0.1000000000D1 * t176 *
#t337
        t346 = 0.5178500000D1 * t319 + 0.36231D1 * rsrhob +
0.1320390000
        #D1 * t322 + 0.993420D0 * t324
        t347 = t346 * t196
        cg17 = 0.375735750D-1 * rsrhob * t50 - 0.9999999999D0 * t190 *
t
        #347
        frhob = (0.4D1 / 0.3D1 * t57 * cg76 - 0.4D1 / 0.3D1 * t60 *
cg76
        #) * dble(t55)
        t354 = cg17 * f
        t356 = cg19 * frhob
        t358 = t209 * cg76
        t360 = 0.4D1 * t63 * t358
        t361 = cg65 - cg93
        t362 = t361 * f
        t364 = t70 * frhob
        t366 = t208 * cg76
        t368 = 0.4D1 * t71 * t366
        cg18 = cg93 + t354 * t68 + t356 * t68 - t360 + t362 * t66 +
t364
        # * t66 + t368
        cg59 = t221 * cg76 / 0.3D1 - t223 * cg76 / 0.3D1
        cg60 = cg66
        cg67 = t230 * cg60 * t7
        trhob = -t233 * t87 * cg59 / 0.2D1 - t85 * t238 * cg67 / 0.2D1
-
        # t242 / 0.2D1
        t378 = cg18 * t89
        t380 = t250 * cg59

```

```

t383 = -t378 * t93 + 0.3D1 * t90 * t380
Arhob = -0.66725D-1 * t246 * t383 * t95
t387 = t113 * cg59
t390 = t109 * trhob
t393 = Arhob * t100
t395 = 0.2D1 * t267 * trhob
t396 = t393 + t395
t397 = t396 * t108
t404 = t393 + t395 + 0.2D1 * t277 * Arhob + 0.4D1 * t281 *
trhob
t408 = 0.133450D0 * t262 * t390 + 0.66725D-1 * t101 * t397 -
0.6
#6725D-1 * t101 * t276 * t404
t409 = t408 * t289
cg22 = cg18 + 0.3D1 * t258 * t387 + t99 * t409
t412 = t130 ** 2
cg36 = dble(t3) / t412 * t75 / 0.3D1
cg28 = -0.3D1 / 0.4D1 * t7 * cg36
t418 = cg29 ** 2
t419 = 0.1D1 / t418
t420 = cg1 * t419
t423 = rhob ** 2
t424 = 0.1D1 / t423
cg46 = -t420 * t135 * cg36 / 0.2D1 - t134 * t424 / 0.2D1
t427 = t140 ** 2
t429 = 0.1D1 / t427 * mu
cg23 = 0.2000000001D1 * t429 * cg42 * cg46
t433 = rhob * cg28
exc_rhob = cg8 * cg33 + t433 * cg33 + t143 * cg23 + cg41 + rho
*
# cg22
t436 = t84 * t86
cg31 = t436 * t2 / 0.2D1
t441 = t89 * t280
t442 = A * cg31
t450 = 0.2D1 * t267 * cg31 + 0.4D1 * t281 * cg31
t454 = 0.133450D0 * t262 * t109 * cg31 + 0.133450D0 * t441 *
t44
#2 * t108 - 0.66725D-1 * t101 * t276 * t450
t455 = t454 * t289
cg84 = t99 * t455
exc_norm_drho = rho * cg84
cg12 = t118 * t120 / 0.2D1
cg87 = 0.2000000001D1 * t310 * cg3 * cg12
exc_norm_drhoa = t128 * cg87
cg88 = t133 * t135 / 0.2D1
cg89 = 0.2000000001D1 * t429 * cg42 * cg88

```

```

exc_norm_drhob = t143 * cg89
t463 = 0.1D1 / t144 / rho
t464 = t1 * t463
cg40 = -0.2D1 * t145 + 0.2D1 * t464
t469 = t144 ** 2
cg69 = -dble(t6) / t147 / t8 * t76 / t469 / 0.18D2 + dble(t6) *
#t149 * t463 / 0.6D1
t479 = rsrhoa * t156
t484 = t12 / t155 / t20
t485 = t167 ** 2
t489 = 0.1D1 / t16
t490 = rsrhoa ** 2
t491 = t489 * t490
t493 = t158 * cg69
t496 = t158 * t490
t498 = t13 * cg69
t501 = t164 * cg69
t507 = t155 ** 2
t509 = t12 / t507
t510 = t23 ** 2
t511 = 0.1D1 / t510
cg90 = -0.1328829340D-1 * cg69 * t24 + 0.4274000000D0 * t479 *
t
#169 - 0.2000000000D1 * t484 * t485 * t168 + 0.9999999999D0 * t157
#* (-0.1898925000D1 * t491 + 0.3797850000D1 * t493 + 0.35876D1 *
cg
#69 + 0.1228650000D1 * t496 + 0.2457300000D1 * t498 + 0.9858800D0
*
# t490 + 0.985880D0 * t501) * t168 + 0.1608182432D2 * t509 * t485
*
# t511
cg7 = cg48
t517 = rsrhoa * t175
t522 = t28 / t174 / t33
t523 = t181 ** 2
t538 = t174 ** 2
t540 = t28 / t538
t541 = t36 ** 2
t542 = 0.1D1 / t541
t548 = rsrhoa * t189
t553 = t41 / t188 / t46
t554 = t195 ** 2
t569 = t188 ** 2
t571 = t41 / t569
t572 = t49 ** 2
t573 = 0.1D1 / t572
cg15 = cg54

```

```

t577 = 0.1D1 / t77
t578 = cg38 ** 2
t583 = 0.1D1 / t78
cg53 = (0.4D1 / 0.9D1 * t577 * t578 + 0.4D1 / 0.3D1 * t57 *
cg40
# + 0.4D1 / 0.9D1 * t583 * t578 - 0.4D1 / 0.3D1 * t60 * cg40) *
dbl
#e(t55)
flrhoa = frhoa
t601 = cg15 * f
t604 = cg19 * flrhoa
t607 = t64 * t65
t622 = cg74 - cg7
t629 = t622 * f
t632 = t70 * flrhoa
t641 = -0.4D1 * t63 * t209 * cg40 + (-0.638837320D-2 * cg69 *
t3
#7 + 0.4109600000D0 * t517 * t183 - 0.2000000000D1 * t522 * t523 *
#t182 + 0.1000000000D1 * t176 * (-0.3529725000D1 * t491 +
0.7059450
#000D1 * t493 + 0.61977D1 * cg69 + 0.2524650000D1 * t496 +
0.504930
#0000D1 * t498 + 0.12503400D1 * t490 + 0.1250340D1 * t501) * t182
+
# 0.3216468318D2 * t540 * t523 * t542 - cg90) * f * t66 + t213 *
f1
#rhoa * t66 + 0.4D1 * t214 * t218 + t622 * frhoa * t66 + t70 *
cg53
# * t66 + 0.4D1 * t216 * t218 + 0.4D1 * t629 * t218 + 0.4D1 * t632
#* t218 + 0.12D2 * t71 * t65 * t578 + 0.4D1 * t71 * t208 * cg40
cg49 = cg90 + (0.375735750D-1 * cg69 * t50 - 0.2225000000D0 *
t5
#48 * t197 + 0.2000000000D1 * t553 * t554 * t196 - 0.9999999999D0
*
# t190 * (-0.2589250000D1 * t491 + 0.5178500000D1 * t493 +
0.36231D
#1 * cg69 + 0.6601950000D0 * t496 + 0.1320390000D1 * t498 +
0.99342
#00D0 * t490 + 0.993420D0 * t501) * t196 - 0.2960857464D1 * t571 *
#t554 * t573) * f * t68 + cg54 * flrhoa * t68 - 0.4D1 * t204 *
t210
# + cg15 * frhoa * t68 + cg19 * cg53 * t68 - 0.4D1 * t206 * t210 -
#0.4D1 * t601 * t210 - 0.4D1 * t604 * t210 - 0.12D2 * t63 * t607 *
#t578 + t641
cg16 = cg7 + t601 * t68 + t604 * t68 - t212 + t629 * t66 + t632
#* t66 + t220
t646 = 0.1D1 / t58

```

```

t651 = 0.1D1 / t61
cg79 = -t646 * t578 / 0.9D1 + t221 * cg40 / 0.3D1 - t651 * t578
#/ 0.9D1 - t223 * cg40 / 0.3D1
cg70 = cg57
t659 = t75 ** 2
cg92 = -0.2D1 / 0.9D1 * dble(t3) / t226 / t80 * t659
t663 = 0.1D1 / t83 / t82
t664 = cg66 ** 2
cg30 = cg4
t671 = cg * t93 * t86
t672 = t2 * cg57
t675 = t233 * t237
t681 = t233 * t241 * cg57 / 0.2D1
t685 = t2 * cg4
t691 = t85 / t236 / cg63
t694 = t237 * t145
t697 = t85 * t694 * cg4 / 0.2D1
t708 = t85 * t86 * t463
cg78 = t671 * t672 * cg70 + t675 * t672 * cg30 / 0.2D1 + t681 -
#t233 * t87 * cg79 / 0.2D1 + t675 * t685 * cg70 / 0.2D1 + t691 *
t6
#85 * cg30 + t697 - t85 * t238 * (-t663 * t664 * t76 / 0.2D1 +
t230
# * cg92 * t7) / 0.2D1 + t233 * t241 * cg70 / 0.2D1 + t85 * t694 *
#cg30 / 0.2D1 + t708
t1rhoa = -t233 * t87 * cg70 / 0.2D1 - t85 * t238 * cg30 / 0.2D1
#- t242 / 0.2D1
t716 = t89 / t244 / t96
t717 = t95 ** 2
t718 = t254 * t717
t719 = cg16 * t89
t721 = t250 * cg70
t724 = -t719 * t93 + 0.3D1 * t90 * t721
t735 = 0.1D1 / t249 / phi
t736 = t735 * cg57
cg83 = 0.133450D0 * t716 * t718 * t724 - 0.66725D-1 * t246 *
(-c
#g49 * t89 * t93 + 0.3D1 * t247 * t721 + 0.3D1 * t719 * t251 -
0.12
#D2 * t90 * t736 * cg70 + 0.3D1 * t90 * t250 * cg79) * t95 -
0.6672
#5D-1 * t246 * t254 * t724 * t95
Alrhoa = -0.66725D-1 * t246 * t724 * t95
t754 = cg27 * phi
t761 = Alrhoa * t100
t763 = 0.2D1 * t267 * t1rhoa
t764 = t761 + t763

```

```

t765 = t764 * t108
t772 = t761 + t763 + 0.2D1 * t277 * Alrhoa + 0.4D1 * t281 *
t1rh
#oa
t776 = 0.133450D0 * t262 * t109 * t1rhoa + 0.66725D-1 * t101 *
t
#765 - 0.66725D-1 * t101 * t276 * t772
t777 = t776 * t289
t793 = t262 * t103
t794 = t275 * trhoa
t804 = cg83 * t100
t805 = Arhoa * t
t807 = 0.2D1 * t805 * t1rhoa
t810 = 0.2D1 * Alrhoa * t * trhoa
t813 = 0.2D1 * A * t1rhoa * trhoa
t815 = 0.2D1 * t267 * cg78
t820 = t270 * t275
t824 = t275 * t284
t832 = t101 * t103
t834 = 0.1D1 / t274 / t107
t835 = t834 * t284
t842 = A * t280
t851 = t104 * t100
t857 = t804 + t807 + t810 + t813 + t815 + 0.2D1 * Alrhoa * t105
#* Arhoa + 0.8D1 * t842 * Arhoa * t1rhoa + 0.2D1 * t277 * cg83 +
0.
#8D1 * t842 * trhoa * Alrhoa + 0.12D2 * t851 * trhoa * t1rhoa +
0.4
#D1 * t281 * cg78
t861 = 0.133450D0 * t89 * t1rhoa * t263 + 0.133450D0 * t262 *
t7
#65 * trhoa - 0.133450D0 * t793 * t794 * t772 + 0.133450D0 * t262
*
# t109 * cg78 + 0.133450D0 * t262 * t271 * t1rhoa + 0.66725D-1 *
t1
#01 * (t804 + t807 + t810 + t813 + t815) * t108 - 0.66725D-1 *
t101
# * t820 * t772 - 0.133450D0 * t793 * t824 * t1rhoa - 0.66725D-1 *
#t101 * t764 * t275 * t284 + 0.133450D0 * t832 * t835 * t772 -
0.66
#725D-1 * t101 * t276 * t857
t864 = t112 ** 2
t865 = 0.1D1 / t864
t866 = t288 * t865
cg34 = -0.2D1 / 0.9D1 * dble(t3) / t293 / t114 * t659
cg35 = cg6
t884 = cg62 ** 2

```



```

cg72 = cg75
t897 = mu ** 2
t898 = 0.1D1 / t308 / t125 * t897
t899 = t122 * cg75
cg37 = 0.2000000001D1 * t310 * cg3 * cg72
exc_rhoa_rhoa = cg35 * cg24 + cg68 * cg37 + cg6 * cg24 - 0.3D1
/
rho
#a * cg35 * cg73 + t128 * (-0.9950248765D1 * t898 * t899 * cg72 +
0
#.2000000001D1 * t310 * cg72 * cg75 + 0.2000000001D1 * t310 * cg3
*
# (cg0 / t299 / cg91 * t120 * t884 + t301 * t305 * cg62 - t301 *
t1
#20 * cg34 / 0.2D1 + t119 / t304 / rhoa)) + cg16 + 0.3D1 * t258 *
t
#113 * cg70 + t99 * t777 + cg20 + rho * (cg49 + 0.6D1 * t754 *
t259
# * cg70 + 0.3D1 * t258 * t777 * cg57 + 0.3D1 * t258 * t113 * cg79
#+ 0.3D1 * t258 * t290 * cg70 + t99 * t861 * t289 - t99 * t866 *
t7
#76)
cg5 = 0.2D1 * t464
cg56 = cg69
t925 = rsrhob * t156
t932 = t489 * rsrhoa * rsrhob
t934 = t158 * cg56
t937 = t159 * rsrhob
t939 = t13 * cg56
t941 = rsrhoa * rsrhob
t943 = t164 * cg56
cg81 = -0.1328829340D-1 * cg56 * t24 + 0.2137000000D0 * t479 *
t
#327 + 0.2137000000D0 * t925 * t169 - 0.2000000000D1 * t484 * t169
#* t326 + 0.9999999999D0 * t157 * (-0.1898925000D1 * t932 +
0.37978
#50000D1 * t934 + 0.35876D1 * cg56 + 0.1228650000D1 * t937 +
0.2457
#300000D1 * t939 + 0.9858800D0 * t941 + 0.985880D0 * t943) * t168
+
# 0.1608182432D2 * t509 * t167 * t511 * t326
t957 = rsrhob * t175
t982 = rsrhob * t189
cg52 = (0.4D1 / 0.9D1 * t577 * cg38 * cg76 + 0.4D1 / 0.3D1 *
t57
# * cg5 + 0.4D1 / 0.9D1 * t583 * cg38 * cg76 - 0.4D1 / 0.3D1 * t60

```

```

    #* cg5) * dble(t55)
    t1032 = t65 * cg38 * cg76
    t1061 = -0.4D1 * t63 * t209 * cg5 + (-0.638837320D-2 * cg56 *
t3
    #7 + 0.2054800000D0 * t517 * t337 + 0.2054800000D0 * t957 * t183 -
    #0.2000000000D1 * t522 * t183 * t336 + 0.1000000000D1 * t176 *
(-0.
    #3529725000D1 * t932 + 0.7059450000D1 * t934 + 0.61977D1 * cg56 +
0
    #.2524650000D1 * t937 + 0.5049300000D1 * t939 + 0.12503400D1 *
t941
    # + 0.1250340D1 * t943) * t182 + 0.3216468318D2 * t540 * t181 *
t54
    #2 * t336 - cg81) * f * t66 + t213 * frhob * t66 + 0.4D1 * t214 *
t
    #366 + t361 * frhoa * t66 + t70 * cg52 * t66 + 0.4D1 * t216 * t366
    #+ 0.4D1 * t362 * t218 + 0.4D1 * t364 * t218 + 0.12D2 * t71 *
t1032
    # + 0.4D1 * t71 * t208 * cg5
    cg64 = cg81 + (0.375735750D-1 * cg56 * t50 - 0.1112500000D0 *
t5
    #48 * t347 - 0.1112500000D0 * t982 * t197 + 0.2000000000D1 * t553
*
    # t197 * t346 - 0.9999999999D0 * t190 * (-0.2589250000D1 * t932 +
0
    #.5178500000D1 * t934 + 0.36231D1 * cg56 + 0.6601950000D0 * t937 +
    #0.1320390000D1 * t939 + 0.9934200D0 * t941 + 0.993420D0 * t943) *
    #t196 - 0.2960857464D1 * t571 * t195 * t573 * t346) * f * t68 +
cg5
    #4 * frhob * t68 - 0.4D1 * t204 * t358 + cg17 * frhoa * t68 + cg19
    #* cg52 * t68 - 0.4D1 * t206 * t358 - 0.4D1 * t354 * t210 - 0.4D1
*
    # t356 * t210 - 0.12D2 * t63 * t64 * t1032 + t1061
    cg39 = -t646 * cg38 * cg76 / 0.9D1 + t221 * cg5 / 0.3D1 - t651
*
    # cg38 * cg76 / 0.9D1 - t223 * cg5 / 0.3D1
    cg77 = cg92
    t1096 = t233 * t241 * cg59 / 0.2D1
    t1099 = t85 * t694 * cg67 / 0.2D1
    cg32 = t671 * t672 * cg59 + t675 * t672 * cg67 / 0.2D1 + t681 -
    #t233 * t87 * cg39 / 0.2D1 + t675 * t685 * cg59 / 0.2D1 + t691 *
t6
    #85 * cg67 + t697 - t85 * t238 * (-t663 * cg66 * t76 * cg60 /
0.2D1
    # + t230 * cg77 * t7) / 0.2D1 + t1096 + t1099 + t708
    cg82 = 0.133450D0 * t716 * t718 * t383 - 0.66725D-1 * t246 *
(-c

```

```

#g64 * t89 * t93 + 0.3D1 * t247 * t380 + 0.3D1 * t378 * t251 -
0.12
#D2 * t90 * t736 * cg59 + 0.3D1 * t90 * t250 * cg39) * t95 -
0.6672
#5D-1 * t246 * t254 * t383 * t95
t1150 = cg82 * t100
t1152 = 0.2D1 * t805 * trhob
t1153 = Arhob * t
t1155 = 0.2D1 * t1153 * trhoa
t1158 = 0.2D1 * A * trhob * trhoa
t1160 = 0.2D1 * t267 * cg32
t1171 = t396 * t275
t1194 = t1150 + t1152 + t1155 + t1158 + t1160 + 0.2D1 * Arhob *
#t105 * Arhoa + 0.8D1 * t842 * Arhoa * trhob + 0.2D1 * t277 * cg82
#+ 0.8D1 * t842 * trhoa * Arhob + 0.12D2 * t851 * trhoa * trhob +
0
#.4D1 * t281 * cg32
t1198 = 0.133450D0 * t89 * trhob * t263 + 0.133450D0 * t262 *
t3
#97 * trhoa - 0.133450D0 * t793 * t794 * t404 + 0.133450D0 * t262
*
# t109 * cg32 + 0.133450D0 * t262 * t271 * trhob + 0.66725D-1 *
t10
#1 * (t1150 + t1152 + t1155 + t1158 + t1160) * t108 - 0.66725D-1 *
#t101 * t820 * t404 - 0.133450D0 * t793 * t824 * trhob -
0.66725D-1
# * t101 * t1171 * t284 + 0.133450D0 * t832 * t835 * t404 -
0.66725
#D-1 * t101 * t276 * t1194
exc_rhoa_rhob = cg22 + cg20 + rho * (cg64 + 0.6D1 * t754 * t259
#* cg59 + 0.3D1 * t258 * t409 * cg57 + 0.3D1 * t258 * t113 * cg39
+
# 0.3D1 * t258 * t290 * cg59 + t99 * t1198 * t289 - t99 * t866 *
t4
#08)
cg10 = 0.2D1 * t145 + 0.2D1 * t464
cg21 = cg56
t1209 = t326 ** 2
t1213 = rsrhob ** 2
t1214 = t489 * t1213
t1216 = t158 * cg21
t1219 = t158 * t1213
t1221 = t13 * cg21
t1224 = t164 * cg21
cg50 = -0.1328829340D-1 * cg21 * t24 + 0.4274000000D0 * t925 *
t
#327 - 0.2000000000D1 * t484 * t1209 * t168 + 0.9999999999D0 *

```

```

t157
# * (-0.1898925000D1 * t1214 + 0.3797850000D1 * t1216 + 0.35876D1
*
# cg21 + 0.1228650000D1 * t1219 + 0.2457300000D1 * t1221 +
0.985880
#0D0 * t1213 + 0.985880D0 * t1224) * t168 + 0.1608182432D2 * t509
*
# t1209 * t511
cg71 = cg93
t1237 = t336 ** 2
t1259 = t346 ** 2
cg14 = cg17
t1277 = cg76 ** 2
cg61 = (0.4D1 / 0.9D1 * t577 * t1277 + 0.4D1 / 0.3D1 * t57 *
cg1
#0 + 0.4D1 / 0.9D1 * t583 * t1277 - 0.4D1 / 0.3D1 * t60 * cg10) *
d
#ble(t55)
flrhob = frhob
t1299 = cg14 * f
t1302 = cg19 * flrhob
t1319 = cg65 - cg71
t1326 = t1319 * f
t1329 = t70 * flrhob
t1338 = -0.4D1 * t63 * t209 * cg10 + (-0.638837320D-2 * cg21 *
t
#37 + 0.4109600000D0 * t957 * t337 - 0.2000000000D1 * t522 * t1237
#* t182 + 0.1000000000D1 * t176 * (-0.3529725000D1 * t1214 +
0.7059
#450000D1 * t1216 + 0.61977D1 * cg21 + 0.2524650000D1 * t1219 +
0.5
#049300000D1 * t1221 + 0.12503400D1 * t1213 + 0.1250340D1 * t1224)
#* t182 + 0.3216468318D2 * t540 * t1237 * t542 - cg50) * f * t66 +
#t361 * flrhob * t66 + 0.4D1 * t362 * t366 + t1319 * frhob * t66 +
#t70 * cg61 * t66 + 0.4D1 * t364 * t366 + 0.4D1 * t1326 * t366 +
0.
#4D1 * t1329 * t366 + 0.12D2 * t71 * t65 * t1277 + 0.4D1 * t71 *
t2
#08 * cg10
cg58 = cg50 + (0.375735750D-1 * cg21 * t50 - 0.2225000000D0 *
t9
#82 * t347 + 0.2000000000D1 * t553 * t1259 * t196 - 0.9999999999D0
#* t190 * (-0.2589250000D1 * t1214 + 0.5178500000D1 * t1216 +
0.362
#31D1 * cg21 + 0.6601950000D0 * t1219 + 0.1320390000D1 * t1221 +
0.
#9934200D0 * t1213 + 0.993420D0 * t1224) * t196 - 0.2960857464D1 *

```

```

#t571 * t1259 * t573) * f * t68 + cg17 * flrhob * t68 - 0.4D1 *
t35
#4 * t358 + cg14 * frhob * t68 + cg19 * cg61 * t68 - 0.4D1 * t356
*
# t358 - 0.4D1 * t1299 * t358 - 0.4D1 * t1302 * t358 - 0.12D2 *
t63
# * t607 * t1277 + t1338
cg85 = cg71 + t1299 * t68 + t1302 * t68 - t360 + t1326 * t66 +
t
#1329 * t66 + t368
cg25 = -t646 * t1277 / 0.9D1 + t221 * cg10 / 0.3D1 - t651 *
t127
#7 / 0.9D1 - t223 * cg10 / 0.3D1
cg43 = cg59
t1351 = cg60 ** 2
cg26 = cg67
t1357 = t2 * cg59
t1366 = t2 * cg67
cg9 = t671 * t1357 * cg43 + t675 * t1357 * cg26 / 0.2D1 + t1096
#- t233 * t87 * cg25 / 0.2D1 + t675 * t1366 * cg43 / 0.2D1 + t691
*
# t1366 * cg26 + t1099 - t85 * t238 * (-t663 * t1351 * t76 / 0.2D1
#+ t230 * cg77 * t7) / 0.2D1 + t233 * t241 * cg43 / 0.2D1 + t85 *
t
#694 * cg26 / 0.2D1 + t708
t1rhob = -t233 * t87 * cg43 / 0.2D1 - t85 * t238 * cg26 / 0.2D1
#- t242 / 0.2D1
t1387 = cg85 * t89
t1389 = t250 * cg43
t1392 = -t1387 * t93 + 0.3D1 * t90 * t1389
cg55 = 0.133450D0 * t716 * t383 * t717 * t1392 - 0.66725D-1 *
t2
#46 * (-cg58 * t89 * t93 + 0.3D1 * t378 * t1389 + 0.3D1 * t1387 *
t
#380 - 0.12D2 * t90 * t735 * cg59 * cg43 + 0.3D1 * t90 * t250 *
cg2
#5) * t95 - 0.66725D-1 * t246 * t383 * t1392 * t95
Alrhob = -0.66725D-1 * t246 * t1392 * t95
t1426 = Alrhob * t100
t1428 = 0.2D1 * t267 * t1rhob
t1429 = t1426 + t1428
t1430 = t1429 * t108
t1437 = t1426 + t1428 + 0.2D1 * t277 * Alrhob + 0.4D1 * t281 *
t
#1rhob
t1441 = 0.133450D0 * t262 * t109 * t1rhob + 0.66725D-1 * t101 *
#t1430 - 0.66725D-1 * t101 * t276 * t1437

```

```

t1442 = t1441 * t289
t1458 = t275 * trhob
t1468 = cg55 * t100
t1470 = 0.2D1 * t1153 * t1rhob
t1473 = 0.2D1 * Alrhob * t * trhob
t1476 = 0.2D1 * A * t1rhob * trhob
t1478 = 0.2D1 * t267 * cg9
t1486 = t275 * t404
t1494 = t834 * t404
t1514 = t1468 + t1470 + t1473 + t1476 + t1478 + 0.2D1 * Alrhob
*
# t105 * Arhob + 0.8D1 * t842 * Arhob * t1rhob + 0.2D1 * t277 *
cg5
#5 + 0.8D1 * t842 * trhob * Alrhob + 0.12D2 * t851 * trhob *
t1rhob
# + 0.4D1 * t281 * cg9
t1518 = 0.133450D0 * t89 * t1rhob * t390 + 0.133450D0 * t262 *
t
#1430 * trhob - 0.133450D0 * t793 * t1458 * t1437 + 0.133450D0 *
t2
#62 * t109 * cg9 + 0.133450D0 * t262 * t397 * t1rhob + 0.66725D-1
*
# t101 * (t1468 + t1470 + t1473 + t1476 + t1478) * t108 -
0.66725D-
#1 * t101 * t1171 * t1437 - 0.133450D0 * t793 * t1486 * t1rhob -
0.
#66725D-1 * t101 * t1429 * t275 * t404 + 0.133450D0 * t832 * t1494
#* t1437 - 0.66725D-1 * t101 * t276 * t1514
t1521 = t408 * t865
cg2 = -0.2D1 / 0.9D1 * dble(t3) / t412 / t129 * t659
cg80 = cg28
t1539 = cg36 ** 2
cg45 = cg46
t1552 = 0.1D1 / t427 / t140 * t897
t1553 = t137 * cg46
cg44 = 0.2000000001D1 * t429 * cg42 * cg45
exc_rhob_rhob = cg80 * cg33 + cg8 * cg44 + cg28 * cg33 - 0.3D1
/
# 0.4D1 * rhob * cg2 * t7 * cg33 + t433 * cg44 + cg8 * cg23 + rhob
#* cg80 * cg23 + t143 * (-0.9950248765D1 * t1552 * t1553 * cg45 +
0
#.2000000001D1 * t429 * cg45 * cg46 + 0.2000000001D1 * t429 * cg42
#* (cg1 / t418 / cg29 * t135 * t1539 + t420 * t424 * cg36 - t420 *
#t135 * cg2 / 0.2D1 + t134 / t423 / rhob)) + cg85 + 0.3D1 * t258 *
#t113 * cg43 + t99 * t1442 + cg22 + rho * (cg58 + 0.6D1 * t754 *
t3
#87 * cg43 + 0.3D1 * t258 * t1442 * cg59 + 0.3D1 * t258 * t113 *

```

```

cg
#25 + 0.3D1 * t258 * t409 * cg43 + t99 * t1518 * t289 - t99 *
t1521
# * t1441)
t1575 = t232 * t86
t1577 = t84 * t237
t1579 = t436 * t145
cg51 = -t1575 * t672 / 0.2D1 - t1577 * t685 / 0.2D1 - t1579 /
0.
#2D1
t1584 = t89 * cg31
t1587 = t101 * A
t1588 = cg31 * t108
t1601 = t805 * cg31
t1602 = t442 * trhoa
t1603 = t267 * cg51
t1612 = t824 * cg31
t1615 = t441 * A
t1636 = 0.133450D0 * t1584 * t263 + 0.266900D0 * t1587 * t1588
*
# trhoa - 0.133450D0 * t793 * t794 * t450 + 0.133450D0 * t262 *
t10
#9 * cg51 + 0.133450D0 * t262 * t271 * cg31 + 0.66725D-1 * t101 *
(
#0.2D1 * t1601 + 0.2D1 * t1602 + 0.2D1 * t1603) * t108 -
0.66725D-1
# * t101 * t820 * t450 - 0.133450D0 * t793 * t1612 - 0.133450D0 *
t
#1615 * t1612 + 0.133450D0 * t832 * t835 * t450 - 0.66725D-1 *
t101
# * t276 * (0.2D1 * t1601 + 0.2D1 * t1602 + 0.2D1 * t1603 + 0.8D1
*
# t842 * Arhoa * cg31 + 0.12D2 * t851 * trhoa * cg31 + 0.4D1 *
t281
# * cg51)
exc_rhoa_norm_drho = cg84 + rho * (0.3D1 * t258 * t455 * cg57 +
#t99 * t1636 * t289 - t99 * t866 * t454)
cg86 = -t1575 * t1357 / 0.2D1 - t1577 * t1366 / 0.2D1 - t1579 /
#0.2D1
t1662 = t1153 * cg31
t1663 = t442 * trhob
t1664 = t267 * cg86
t1673 = t1486 * cg31
t1696 = 0.133450D0 * t1584 * t390 + 0.266900D0 * t1587 * t1588
*
# trhob - 0.133450D0 * t793 * t1458 * t450 + 0.133450D0 * t262 *
t1

```

```

#09 * cg86 + 0.133450D0 * t262 * t397 * cg31 + 0.66725D-1 * t101 *
#(0.2D1 * t1662 + 0.2D1 * t1663 + 0.2D1 * t1664) * t108 -
0.66725D-
#1 * t101 * t1171 * t450 - 0.133450D0 * t793 * t1673 - 0.133450D0
*
# t1615 * t1673 + 0.133450D0 * t832 * t1494 * t450 - 0.66725D-1 *
t
#101 * t276 * (0.2D1 * t1662 + 0.2D1 * t1663 + 0.2D1 * t1664 +
0.8D
#1 * t842 * Arhob * cg31 + 0.12D2 * t851 * trhob * cg31 + 0.4D1 *
t
#281 * cg86)
exc_rhob_norm_drho = cg84 + rho * (0.3D1 * t258 * t455 * cg59 +
#t99 * t1696 * t289 - t99 * t1521 * t454)
t1702 = cg31 ** 2
t1706 = A * t1702
t1711 = t275 * cg31 * t450
t1717 = t450 ** 2
t1731 = t454 ** 2
exc_norm_drho_norm_drho = rho * (t99 * (0.133450D0 * t89 *
t1702
# * t109 + 0.667250D0 * t101 * t1706 * t108 - 0.266900D0 * t793 *
t
#1711 - 0.266900D0 * t1615 * t1711 + 0.133450D0 * t101 * t103 *
t83
#4 * t1717 - 0.66725D-1 * t101 * t276 * (0.2D1 * t1706 + 0.12D2 *
t
#851 * t1702)) * t289 - t99 * t1731 * t865)
exc_rhoa_norm_drhoa = cg68 * cg87 + t314 * cg87 + t128 *
(-0.995
#0248765D1 * t898 * t899 * cg12 + 0.2000000001D1 * t310 * cg12 *
cg
#75 + 0.2000000001D1 * t310 * cg3 * (-t300 * t120 * cg62 / 0.2D1 -
#t118 * t305 / 0.2D1))
t1750 = cg12 ** 2
exc_norm_drhoa_norm_drhoa = t128 * (-0.9950248765D1 * t898 *
t12
#2 * t1750 + 0.2000000001D1 * t310 * t1750)
exc_rhob_norm_drhob = cg8 * cg89 + t433 * cg89 + t143 *
(-0.9950
#248765D1 * t1552 * t1553 * cg88 + 0.2000000001D1 * t429 * cg88 *
c
#g46 + 0.2000000001D1 * t429 * cg42 * (-t419 * t135 * cg36 / 0.2D1
#- t133 * t424 / 0.2D1))
t1772 = cg88 ** 2
exc_norm_drhob_norm_drhob = t143 * (-0.9950248765D1 * t1552 *
t1

```



```

#37 * t1772 + 0.20000000001D1 * t429 * t1772)
cg94 = exc_norm_drhob_norm_drhob
return
end

```

>

```

> evalf(r_eqs_lsd4(0.1,0.1,0.3,0.15,0.15));
evalf(exc);
-0.1084516963
-0.09857616110

```

```

> evalf(r_eqs_lda4(0.2,0.3));
evalf(exc);
-0.1000665799
-0.09857616110

```

```

> exc:='exc':exc_rhoa:='exc_rhoa':exc_rhob:='exc_rhob':exc_norm_drho:='exc_norm_drho':
exc_norm_drhoa:='exc_norm_drhoa':exc_norm_drhob:='exc_norm_drhob':
exc_rhoa_rhoa:='exc_rhoa_rhoa':exc_rhoa_rhob:='exc_rhoa_rhob':exc_rhob_rho
b:='exc_rhob_rhob':
exc_rhoa_norm_drho:='exc_rhoa_norm_drho':exc_rhob_norm_drho:='exc_rhob_norm_drho':
exc_norm_drho_norm_drho:='exc_norm_drho_norm_drho':
exc_rhoa_norm_drhoa:='exc_rhoa_norm_drhoa':exc_norm_drhoa_norm_drhoa:='exc_norm_drhoa_norm_drhoa':
exc_rhob_norm_drhob:='exc_rhob_norm_drhob':exc_norm_drhob_norm_drhob:='exc_norm_drhob_norm_drhob':
my_rho:='my_rho':my_norm_drho:='my_norm_drho':my_rhoa:='my_rhoa':my_rhob:='my_rhob':
my_norm_drhoa:='my_norm_drhoa':my_norm_drhob:='my_norm_drhob':

exc_rho:='exc_rho':exc_rho_rho:='exc_rho_rho':exc_rho_norm_drho:='exc_rho_norm_drho':

```

>