流体解析プログラムの並列化

報告書

平成２１年３月

Advanced Algorithm & Systems

目　次

ページ

１．　概要

流体解析プログラムを対象として、OpenMP を用い、マルチコアマシン向け並列化プログラムへの修正を行った。

プログラムは以下の特徴を有している。

（1）　独自 SGS モデルによる LES（Large Eddy Simulation）ソルバ
（2）　重合格子（複数の構造格子）
（3）　マトリックスソルバとしてマルチカラーSOR 法を使用

行った具体的な作業は、以下の通りである。

（１）指示行の挿入
（２）必要に応じてのプログラム変更
（３）検証計算

作業を以下の手順で行った。

（1）　提供プログラム（prog00）、提供データ（data250）を OpenMP オプションを使用しないで実行し、提供結果（result250）と比較し、マシンの違いによって結果が一致することを確認した。
（2）　提供プログラム（prog00）、提供データ（data250）を OpenMP オプションを使用して実行し、提供結果（result250）と比較し、コンパイラオプションによらず結果が一致することを確認した。
（3）　並列化対象となるサブルーチンを一つずつ独立に並列化し、２ステップについて並列化しない場合の結果と一致することを確認した。
（4）　並列化されたサブルーチンをすべて使用し、250 ステップについて並列化しない場合の結果と比較し、完全に一致することを確認した。

以下の開発環境で作業を行った。

１．コンパイラ：　GNU　gfortran
２．　機種：powerEdge 1600SC
３．　CPU：XEON　2.8GHz　x　４
４．　メモリ：　2GB
５．　OS：　FedoraCore６

２．　並列化対象ファイル

重要度の高いサブルーチン群（カテゴリー１）と比較的重要度の高いサブルーチン群（カテゴリー２）について OpenMp による並列化を行った。サブルーチンが含まれているファイル名は以下の通りである。

並列化コードについて指示行の位置とコメントと添付資料Aとしてまとめた。

表１　並列化対象ファイル

	整理番号	ファイル名（*.f）
カテゴリー１	1	sor7p
	2	sor7p_SW
	3	residp
	4	sor7v
	5	sor7v_SW
	6	sgs
	7	sgsbnd
	8	corvcn
	9	corbnd
	10	rhspc
	11	smacco
	12	smcbnd
カテゴリー２	13	change1
	14	change2
	15	filter
	16	shear
	17	dyn
	18	sgscal
	19	calc_grad_p
	20	intp_2
	21	intv_2
	22	wallf0
	23	lawall
	24	cdsl
	25	favr19
	26	ucdiff
	27	dampfm

３．　検証結果

3.1　単独に並列化を行った場合の cpu

各ファイルごとに並列化を行い、そのファイルに含まれるサブルーチンのみを並列化として扱い、他のサブルーチンは並列化しないで検証を行った。実行は２ステップである。

次ページの表３に、標準出力の最後にある「+++ last real cpu time」の cpu をまとめた。

3.2　対象ファイルをすべて並列化を行った場合の cpu

以下に対象としたすべてのサブルーチンを並列化して１５ステップを計算した場合の結果を示す。完全に一致することを確認した。

表２　結果の比較

		並列化あり	並列化なし
cpu		4363.867	2985.077
div	vol-1	2.491D-02	2.491D-02
.	vol-2	1.003D-05	1.003D-05
rel.res	vol-1	6.885D-04	6.885D-04
	vol-2	1.699D-04	1.699D-04
bp	vol-1	3.697D+00	3.697D+00
	vol-2	2.663D+00	2.663D+00
abs.res	vol-1	2.545D-03	2.545D-03
	vol-2	4.525D-04	4.525D-04

表２　サブルーチンを単独で並列化した場合の cpu（秒）注）

		ファイル名	並列化した場合
カテゴリー１	1	sor7p	643.2432
	2	sor7p_SW	1087.263
	3	residp	428.9918
	4	sor7v	431.1544
	5	sor7v_SW	497.7703
	6	sgs	448.0169
	7	sgsbnd	430.6275
	8	corvcn	425.7203
	9	corbnd	432.0903
	10	rhspc	425.2614
	11	smacco	448.7408
	12	smcbnd	426.3892
カテゴリー２	13	change1	448.7408
	14	change2	431.2694
	15	filter	424.4934
	16	shear	426.6682
	17	dyn	426.6901
	18	sgscal	428.6818
	19	calc_grad_p	427.7390
	20	intp_2	428.0349
	21	intv_2	428.1259
	22	wallf0	431.9923
	23	lawall	430.9645
	24	cdsl	433.8080
	25	favr19	429.9836
	26	ucdiff	428.5748
	27	dampfm	428.6078

並列化をまったくしない場合の cpu　＝　428.4529　秒

注）4cpu を合わせた時間を掲載しているので、並列化なしと比較するには上記表中時間の 1/4 とする必要がある。

４． まとめ

以下に作業から得られた知見をまとめた。

１． 並列化によって解の精度がまったく保証される。

２． 並列化によって約２倍の高速化が図られた。

３． コードを書き換えることによって更に効率化することが可能である。

添付資料 A

並列化の詳細

　本文中、表１に示すファイルにコーディングされているサブルーチンのそれぞれについて、並列化した位置を以下に示す。また、並列化にあたっては、一部修正した箇所がある。

　並列化は重要と思われる do ループについて行ったが、他の do ループを並列化できる可能性もあるために、サブルーチン内のすべての do ループも合わせて記載した。

A1　sor7p.f

（1）並列化ループ

```
      subroutine sor7p(p0, m1max)

        do 500 itrip=1,litrip

c$omp parallel
c$omp&  default(private)
c$omp&  shared(p0,m1max, litrip,nvolu,
c$omp&         nx,ny,nz,istr,iend,jstr,jend,
c$omp&         kstr,n1ds,itype,jtype,ktype,
c$omp&         init0,icon,difal,sumall,difsqr,sumsqr,
c$omp&         epsden,epsdif,
c$omp&         coord_type,n1rs,n1re,div,rsqrp,rbps,
c$omp&         epspa,epspr,iterp,litrp,itrsp
c$omp&         )
c$omp do reduction(+ : difall,sumall,icon)
          do 505 m=1,nvolu
  505     continue
c$omp end do nowait
c$omp end parallel

          if(icon.eq.0.and.itrip.gt.1) go to 800

  500   continue

  800   continue
```

（2）その他の可能性と問題点

do 500 itrip=1,litrip について並列化も考えられるが、その場合は、go to 800 によって do 500 から抜け出さなくともよいコードに変更する必要がある。

A 2　sor7p_SW.f

（1）並列化ループ

```
      subroutine sor7p_B(p0, m1max, m, iip1, ijp1)

              do 110 ic=0,6
c
c$omp parallel
c$omp&  default(private)
c$omp&  shared(p0,m1max,m, iip1, ijp1,
c$omp&         n1rs,n1re, ic,
c$omp&         c1p,omegap,div,bp   )
c$omp do
                do 100 n1=n1rs(m)+ic,n1re(m),7

  100           continue
c$omp end do nowait
c$omp end parallel

  110           continue


      subroutine sor7p_D(p0, m1max, m, iip1, ijp1)

              do 110 ic=0,6

c$omp parallel
c$omp&  default(private)
c$omp&  shared(po,m1max,m, iip1, ijp1,
c$omp&         n1rs,n1re,ic,
c$omp&         c1p,omegap,div   )

c$omp do
                do 100 n1=n1rs(m)+ic,n1re(m),7

  100           continue
c$omp end do nowait
c$omp end parallel

  110           continue
```

（2）その他の可能性と問題点

do 110 を並列化することも考えられるが試していない。

A 3　residp.f

（1）　並列化ループ

```
      subroutine residp(p0, m1max, m)

c$omp parallel
c$omp&  default(private)
c$omp&  shared(p0,m1max,m,iip1,ijp1,rsqrs,
c$omp&         bp,n1rs,n1re,ipflag,div,clp
c$omp&         )
c$omp do reduction(+ : rsqrps)
      do 100 n1=n1rs(m),n1re(m)

  100 continue
c$omp end do nowait
c$omp end parallel
```

（2）その他の可能性と問題点

このサブルーチンについては do 100 のみであるので、他の可能性はない。

A 4 sor7v.f

（1）並列化ループ

```
      subroutine sor7v(m1max)

      do 10 m=1,nvol
        do 15 n1=n1rs(m),n1re(m)
   15   continue
   10 continue
c
      do 17 n1=n1rs(1),n1re(nvol)
   17 continue
        do 500 itriv=1,litriv
cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m1max,nvol,icon,icono2,
cc$omp&         literv,
cc$omp&         nx,ny,nz,istr,iend,jstr,jend,kstr,n1ds,
cc$omp&         itype,jtype,ktype,
cc$omp&         iwzero,epsva,epsvr,
cc$omp&         coord_type,bw2,w,iterv,itrsv,
cc$omp&         rbus,rbvs,rbws,busqr,bvsqr,bwsqr   )
cc$omp do
          do 505 m=1,nvol
               iskip = 0
cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(iskip,m,m1max,nvol,icon,icon2,
cc$omp&         literv,nxm0, nym0, nzm0,
cc$omp&         iip1,ijp1,n1dstr,iip2,ijp2,ist,jst,kst,
cc$omp&         ktypn,ktypm,jtypn,jtypm,itypn,itypm,
cc$omp&         iwzero,epsva,epsvr,
cc$omp&         coord_type,bw2,w,iterv,itrsv,
cc$omp&         rbus,rbvs,rbws,busqr,bvsqr,bwsqr   )
cc*aas<-
cc$omp do
            do 510 itrv=1,literv
c
c*aas->
               if(iskip .eq. 1) goto 519
c*aas<-
              do 16 k=1,nzm0-1
                do 26 j=1,nym0-1
                do 26 i=1,nxm0-1
   26           continue
   16         continue
c
              do 11 k=1,nzm0-1
                do 21 j=1,nym0-1
                do 21 i=1,nxm0-1
   21           continue
   11         continue
              do 19 j=1,nym0-1
              do 19 i=1,nxm0-1
   19         continue
c*ame->
```

```
        iskip = 1
c*ame<-
c*original          go to 511
c
 519  continue  !*aas
 510  continue
cc$omp end do nowait
cc$omp end parallel

c*ame->
      if(iskip .eq. 0) then
      end if
c*ame<-
 511  continue

 505     continue
cc$omp end do nowait
cc$omp end parallel

     if(icon.eq.0.and.icono2.eq.0) go to 800

 500    continue

 800    continue


    do 540 m=1,nvol

c$omp parallel
c$omp&  default(private)
c$omp&  shared(m1max,m,nvol,icon,icono2,
c$omp&        iipl,ijpl,iip2,ijp2,
c$omp&        nlrs,nlre,u,v,w,clp,
c$omp&        bu,bv,bw,bw2,div,
c$omp&        rsqr1m,rsqr2m,rsqr3m
c$omp&        )
c$omp do reduction(+ : rsqr1m,rsqr2m,rsqr3m)
     do 101 nl=nlrs(m),nlre(m)
 101     continue
c$omp end do nowait
c$omp end parallel

 540    continue

    do 200 nl=nlrs(m),nlre(m)
 200    continue

c
```

（２）その他の可能性と問題点

do 500 itriv=1,litriv、do 505 m=1,nvol および do 510 itriv=1,litriv の並列化も可能である。
do 505 m=1,nvol の並列化が一番やりやすい。
do 500 itriv=1,litriv を並列化するには、goto800 の処理を書き換える必要がある。
また do 510 itriv=1,litriv の場合、go to 511 を iskip という変数を用いて書き換えることも可能である。
　その他の do ループは、小さいのであまり並列化する利点はないように思われる。

A 5　sor7v_SW.f

（1）　並列化ループ

```
      subroutine sor7v_B(bw2, m1max, m, iip1, ijp1, k2d,
     &                   rsqr1m, rsqr2m, rsqr3m)

              do 110 ic=0,6
c$omp parallel
c$omp&  default(private)
c$omp&  shared(bw2, m1max, m, iip1, ijp1, k2d,
c$omp&         rsqr1m, rsqr2m, rsqr3m,ic,n1rs,n1re,
c$omp&         omegav,clp,
c$omp&         u,v,w,bu,bv,bw )
c$omp do reduction(+ : rsqr1m, rsqr2m, rsqr3m)
                do 100 n1=n1rs(m)+ic,n1re(m),7
  100           continue
c$omp end do nowait
c$omp end parallel
  110         continue

c
************************************************************************
      subroutine sor7v_D(bw2, m1max, m, iip1, ijp1, k2d,
     &                   rsqr1m, rsqr2m, rsqr3m)
c
              do 110 ic=0,6
c$omp parallel
c$omp&  default(private)
c$omp&  shared(bw2, m1max, m, iip1, ijp1, k2d,
c$omp&         rsqr1m, rsqr2m, rsqr3m,ic,n1rs,n1re,
c$omp&         omegav,clp,
c$omp&         u,v,w,bu,bv,bw )
c$omp do reduction(+ : rsqr1m, rsqr2m, rsqr3m)
                do 100 n1=n1rs(m)+ic,n1re(m),7
                  rsqr1m=rsqr1m + resu*resu
                  rsqr2m=rsqr2m + resv*resv
                  rsqr3m=rsqr3m + resw*resw
c*original                  rsqr1m=resu*resu+rsqr1m
c*original                  rsqr2m=resv*resv+rsqr2m
c*original                  rsqr3m=resw*resw+rsqr3m
cc
  100           continue
c$omp end do nowait
c$omp end parallel
  110         continue
```

（2）その他の可能性と問題点

他の並列化の可能性はない。

A 6　sgs.f

（1）並列化ループ

```
      subroutine sgs

      do 140 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(itpara,itparas,itpara2,itflag,aplseg,raplseg,
c$omp&         cdes,rsq_cdes,csgs,yepower_S,
c$omp&         m,n1rs,n1re,iip1,ijp1,
c$omp&         iflag,jflag,kflag,
c$omp&         gg,u,v,w,rho,damp,amu,idfsgs,distbl,apls,
c$omp&         ajac,dles,akappa,
c$omp&         ipflag,idsgs,imtsd,
c$omp&         tk,tkes,yeps,blend,tesgs,iblend,
c$omp&         rtspara_rans,dzero,cmyu_rans,timescale,
c$omp&         cdyn,amu_t,amu_k,te,
c$omp&         tqsgs,rtspara,rsige,ceta,prl,
c$omp&         tdyn,ihmtfl,alf_t,
c$omp&         icfl_fm, dfm_t,rschfmt, work
c$omp&         )
c$omp do
        do 230 n1=n1rs(m),n1re(m)
  230 continue
c$omp end do nowait
c$omp end parallel
c
  140 continue
```

（2）その他の可能性と問題点

他の並列化の可能性はない。

A 7 sgsbnd.f

（1）並列化ループ

```
      subroutine sgsbnd

c$omp parallel
c$omp&  default(private)
c$omp&  shared(alps,akappa,alamda,yepower_S,
c$omp&         itparas,itpara,itpara2,itflag,raplseg,
c$omp&         cdes,rsq_cdes,iuwall,ibwf,
c$omp&         n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         iflst,jflst,kflst,imlst,
c$omp&         gg,n1lst,n1bnd,uvwbnd,tebnd,u,v,w,rho,
c$omp&         damp,amu,idfsgs,distbl,ajac,dles,imtsd,csgs,
c$omp&         iflag,tk,tkes,yeps,blend,
c$omp&         iblend,rtspara_rans,cmyu_rans,dzero,rstpara,
c$omp&         cdyn,amu_t,amu_k,te,tqsgs,rsige,ceta,prl,
c$omp&         tdyn,alf_t,ihmtfl,dfm_t,rschfmt
c$omp&         )
c$omp do
      do 140 n1b=1,n1bmax
  140 continue
c$omp end do nowait
c$omp end parallel

c$omp parallel
c$omp&  default(private)
c$omp&  shared(alps,akappa,alamda,yepower_S,
c$omp&         itparas,itpara,itpara2,itflag,raplseg,
c$omp&         cdes,rsq_cdes,iuwall,ibwf,
c$omp&         n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         iflst,jflst,kflst,imlst,
c$omp&         gg,n1lst,n1bnd,uvwbnd,tebnd,u,v,w,rho,
c$omp&         damp,amu,idfsgs,distbl,ajac,dles,imtsd,csgs,
c$omp&         iflag,tk,tkes,yeps,blend,
c$omp&         iblend,rtspara_rans,cmyu_rans,dzero,rstpara,
c$omp&         cdyn,amu_t,amu_k,te,tqsgs,rsige,ceta,prl,
c$omp&         tdyn,alf_t,ihmtfl,dfm_t,rschfmt
c$omp&         )
c$omp do
      do 240 n1b=1,n1bmax
  240 continue
c$omp end do nowait
c$omp end parallel

c$omp parallel
c$omp&  default(private)
c$omp&  shared(alps,akappa,alamda,yepower_S,
c$omp&         itparas,itpara,itpara2,itflag,raplseg,
c$omp&         cdes,rsq_cdes,iuwall,ibwf,
c$omp&         n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         iflst,jflst,kflst,imlst,
c$omp&         gg,n1lst,n1bnd,uvwbnd,tebnd,u,v,w,rho,
c$omp&         damp,amu,idfsgs,distbl,ajac,dles,imtsd,csgs,
c$omp&         iflag,tk,tkes,yeps,blend,
c$omp&         iblend,rtspara_rans,cmyu_rans,dzero,rstpara,
c$omp&         cdyn,amu_t,amu_k,te,tqsgs,rsige,ceta,prl,
c$omp&         tdyn,alf_t,ihmtfl,dfm_t,rschfmt,work
c$omp&         )
c$omp do
      do 340 n1b=1,n1bmax

  340 continue
c$omp end do nowait
c$omp end parallel

c$omp parallel
c$omp&  default(private)
c$omp&  shared(alps,akappa,alamda,yepower_S,
c$omp&         itparas,itpara,itpara2,itflag,raplseg,
c$omp&         cdes,rsq_cdes,iuwall,ibwf,
c$omp&         n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         iflst,jflst,kflst,imlst,
c$omp&         gg,n1lst,n1bnd,uvwbnd,tebnd,u,v,w,rho,
c$omp&         damp,amu,idfsgs,distbl,ajac,dles,imtsd,csgs,
c$omp&         iflag,tk,tkes,yeps,blend,
c$omp&         iblend,rtspara_rans,cmyu_rans,dzero,rstpara,
c$omp&         cdyn,amu_t,amu_k,te,tqsgs,rsige,ceta,prl,
c$omp&         tdyn,alf_t,ihmtfl,dfm_t,rschfmt
c$omp&         )
c$omp do
      do 141 n1b=1,n1bmax
  141 continue
c$omp end do nowait
c$omp end parallel

c$omp parallel
c$omp&  default(private)
c$omp&  shared(alps,akappa,alamda,yepower_S,
c$omp&         itparas,itpara,itpara2,itflag,raplseg,
c$omp&         cdes,rsq_cdes,iuwall,ibwf,
c$omp&         n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         iflst,jflst,kflst,imlst,
c$omp&         gg,n1lst,n1bnd,uvwbnd,tebnd,u,v,w,rho,
c$omp&         damp,amu,idfsgs,distbl,ajac,dles,imtsd,csgs,
c$omp&         iflag,tk,tkes,yeps,blend,
c$omp&         iblend,rtspara_rans,cmyu_rans,dzero,rstpara,
c$omp&         cdyn,amu_t,amu_k,te,tqsgs,rsige,ceta,prl,
c$omp&         tdyn,alf_t,ihmtfl,dfm_t,rschfmt
c$omp&         )
c$omp do
      do 241 n1b=1,n1bmax
  241 continue
c$omp end do nowait
c$omp end parallel

c$omp parallel
c$omp&  default(private)
c$omp&  shared(alps,akappa,alamda,yepower_S,
c$omp&         itparas,itpara,itpara2,itflag,raplseg,
c$omp&         cdes,rsq_cdes,iuwall,ibwf,
c$omp&         n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         iflst,jflst,kflst,imlst,
c$omp&         gg,n1lst,n1bnd,uvwbnd,tebnd,u,v,w,rho,
c$omp&         damp,amu,idfsgs,distbl,ajac,dles,imtsd,csgs,
c$omp&         iflag,tk,tkes,yeps,blend,
c$omp&         iblend,rtspara_rans,cmyu_rans,dzero,rstpara,
c$omp&         cdyn,amu_t,amu_k,te,tqsgs,rsige,ceta,prl,
c$omp&         tdyn,alf_t,ihmtfl,dfm_t,rschfmt
c$omp&         )
c$omp do
      do 341 n1b=1,n1bmax
  341 continue
c$omp end do nowait
c$omp end parallel

cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(mvbnd,n1dbnd,iend,istr,jend,jstr,
cc$omp&         amu_t,amu_k,alf_t,dfm_t,iflbnd
cc$omp&         )
cc$omp do
      do 150 n1b1=1,n1b1max
  150 continue
cc$omp end do nowait
cc$omp end parallel

cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(mvbnd,n1dbnd,iend,istr,jend,jstr,
cc$omp&         amu_t,amu_k,alf_t,dfm_t,iflbnd
cc$omp&         )
cc$omp do
      do 152 n1b1=n1b1max+1,n1b2max
  152 continue
cc$omp end do nowait
cc$omp end parallel

cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(mvbnd,n1dbnd,iend,istr,jend,jstr,
cc$omp&         amu_t,amu_k,alf_t,dfm_t,iflbnd
cc$omp&         )
cc$omp do
      do 154 n1b1=n1b2max+1,n1b3max
  154 continue
cc$omp end do nowait
cc$omp end parallel
c
c*AAS: No OMP is used beacuse nvol shoul be in sequence.
c
cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(nx,ny,nz,iend,istr,jend,jstr,n1ds,
cc$omp&         ktype,jtype,itype,amu_t,amu_k,alf_t,dfm_t
cc$omp&         )
cc$omp do
      do 800 m=1,nvol
        do 422 k=0,nzm0
          do 424 j=0,nym0
  424     continue
  422   continue
        do 432 k=0,nzm0
        do 434 i=ist,ist+iip1-1
434     continue
432   continue
      do 442 n1k0=n1st,n1st+ijp1-1
442   continue
800 continue
```

（２）その他の可能性と問題点

do 150、152、154 については、小さいので並列化の利点はなさそうである。do 800 は、変数の置き換えだけなので必要ないと思われる。

A 8 corvcn.f

（1） 並列化ループ

```
      subroutine corvcn

c$omp parallel
c$omp&  default(private)
c$omp&  shared(nvol,n1rs,n1re,istr,iend,jstr,jend,
c$omp&         iflag,jflag,kflag,
c$omp&         amu,amu_t,ajac,cup1v,
c$omp&         gg,gc,uc,vc,wc,u,v,w,
c$omp&         ieno,pcup,clp,bu,bv,bw,
c$omp&         dt,imach,n1lst,kcomp,te,tu,tv,tw,
c$omp&         rho,rochn,cnu,cnv,cnw,rhoold,grav,
c$omp&         grad_p,icorv0,idp,dpdz
c$omp&         )
c$omp do
      do 140 m=1,nvol
cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m,n1rs,n1re,iip1,ijp1,
cc$omp&         iflag,jflag,kflag,
cc$omp&         amu,amu_t,ajac,cup1v,
cc$omp&         gg,gc,uc,vc,wc,u,v,w,
cc$omp&         ieno,pcup,clp,bu,bv,bw
cc$omp&         )
cc$omp do
        do 230 n1=n1rs(m),n1re(m)
  230 continue
cc$omp end do nowait
cc$omp end parallel


cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m,n1rs,n1re,iip1,ijp1,ieno,
cc$omp&         iflag,jflag,kflag,
cc$omp&         amu,ajac,
cc$omp&         xc,yc,zc,gg,gc,uc,u,v,w,pcup,wc,clp,bu,bv,bw
cc$omp&         )
cc$omp do
        do 232 n1=n1rs(m),n1re(m)
  232 continue
cc$omp end do nowait
cc$omp end parallel


cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m,n1rs,n1re,iip1,ijp1,ieno,
cc$omp&         iflag,jflag,kflag,
cc$omp&         amu,ajac,
cc$omp&         xc,yc,zc,gg,gc,uc,u,v,w,pcup,wc,clp,bu,bv,bw
cc$omp&         )
cc$omp do
        do 234 n1=n1rs(m),n1re(m)
  234 continue
cc$omp end do nowait
```

```
cc$omp end parallel


cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m,n1rs,n1re,iip1,ijp1,
cc$omp&         uc,vc,wc,ajac,dt,imach,clp,
cc$omp&         n1lst,kcomp,te,rho,bu,bv,bw,tu,tv,tw,rochn,
cc$omp&         cnu,cnv,cnw,u,v,w,rhoold,idp,grad_p,icorv0,
cc$omp&         grav,dpdz
cc$omp&         )
cc$omp do
       do 240 n1=n1rs(m),n1re(m)
  240 continue
cc$omp end do nowait
cc$omp end parallel



cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m,n1rs,n1re,
cc$omp&         iflag,jflag,kflag,
cc$omp&         clp,u,v,w,bu,bv,bw
cc$omp&         )
cc$omp do
       do 330 n1=n1rs(m),n1re(m)
  330   continue
cc$omp end do nowait
cc$omp end parallel

  140 continue
c$omp end do nowait
c$omp end parallel
```

（２）その他の可能性と問題点

一番外側の do140 について並列化を行った。しかし、内部の do ループで行うことも考えられる。両者の比較は行っていない。

A 9 corbnd.f

```
      subroutine corbnd
c
c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         iflag,jflag,kflag,dt,ajac,iflst,jflst,kflst,
c$omp&         uc,vc,wc,gc,u,v,w,ieno,
c$omp&         pcup,cup1v,clp,bu,bv,bw
c$omp&         )
c$omp do
      do 138 n1b=1,n1bmax
  138 continue
c$omp end do nowait
c$omp end parallel

c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         amu,rrho,amu_t,dt,ajac,iflst,jflst,kflst,imlst,
c$omp&         gc,gg,clp,n1bnd,uvwbnd,u,v,w,bu,bv,bw
c$omp&         )
c$omp do
      do 140 n1b=1,n1bmax
c
  140 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         amu,rrho,amu_t,dt,ajac,iflst,jflst,kflst,imlst,
c$omp&         gc,gg,clp,n1bnd,uvwbnd,u,v,w,bu,bv,bw
c$omp&         )
c$omp do
      do 142 n1b=1,n1bmax

  142 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         amu,rrho,amu_t,dt,ajac,iflst,jflst,kflst,imlst,
c$omp&         gc,gg,clp,n1bnd,uvwbnd,u,v,w,bu,bv,bw
c$omp&         )
c$omp do
      do 144 n1b=1,n1bmax

  144 continue
c$omp end do nowait
c$omp end parallel
```

```
cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
cc$omp&         ipbc,nvdrv0,u,v,w,iflag,jflag,kflag,
cc$omp&         uc,vc,wc,dt,ajac,imach,
cc$omp&         clp,kcomp,te,tu,tv,tw,rho,rochn,rhoold,
cc$omp&         bu,bv,bw,cnu,cnv,cnw,icorv0,idp,grad_p,grav
cc$omp&         )
cc$omp do
      do 150 n1b=1,n1bmax

  150 continue
cc$omp end do nowait
cc$omp end parallel
```

（２）　その他の可能性と問題点

最後の do 150 n1b=1,n1bmax も並列化のメリットの可能性はある。

A 10　rhspc.f

（1）並列化ループ

```
      subroutine rhspc(lcorv)

      do 440 m=1,nvol
c
c$omp parallel
c$omp&  default(private)
c$omp&  shared(lcorv,m,n1rs,n1re,iip1,ijp1,
c$omp&         amach2,p,pold,rdt,abp,bp,uc,vc,wc,ajac,
c$omp&         ipbc,ipflag,clp
c$omp&         )
c$omp do
        do 430 n1=n1rs(m),n1re(m)
  430   continue
c$omp end do nowait
c$omp end parallel

c
  440 continue


      do 140 m=1,nvol
c
c$omp parallel
c$omp&  default(private)
c$omp&  shared(lcorv,m,n1rs,n1re,
c$omp&         iflag,jflag,kflag,ipflag,
c$omp&         bp,p,constp,lcomp2,rho_ref,
c$omp&         grav,xc,yc,zc,clp
c$omp&         )
c$omp do
c
        do 130 n1=n1rs(m),n1re(m)
  130   continue
c$omp end do nowait
c$omp end parallel
c
  140 continue
```

（2）その他の可能性と問題点

外側のループ do 440、do 140 で並列化できる。試していない。

A 11　smacco.f

（1） 並列化ループ

```
      subroutine smacco

      do 30 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1,
c$omp&         iflag,jflag,kflag,ipflag,
c$omp&         xc,yc,zc,gc,gg,
c$omp&         dt,ajac,amach2,rdt,
c$omp&         clp
c$omp&         )
c$omp do
        do 20 n1=n1rs(m),n1re(m)
   20   continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1
c$omp&         ,iflag,jflag,kflag,ipflag
c$omp&         ,xc,yc,zc
c$omp&         ,u,v,w
c$omp&         ,ajac,amach2
c$omp&         ,clp
c$omp&         )
c$omp do
        do 40 n1=n1rs(m),n1re(m)
   40   continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1
c$omp&         ,iflag,jflag,kflag,ipflag
c$omp&         ,xc,yc,zc
c$omp&         ,u,v,w
c$omp&         ,ajac,kcomp,rho,rrho,rhoold
c$omp&         ,bp
c$omp&         )
c$omp do
        do 45 n1=n1rs(m),n1re(m)
   45 continue
c$omp end do nowait
c$omp end parallel

   30 continue

      do 540 m=1,nvol
c$omp parallel
c$omp&  default(private)
```

```
c$omp&  shared(m,n1rs,n1re
c$omp&         ,iflag,jflag,kflag
c$omp&         ,clp
c$omp&         )
c$omp do
        do 530 n1=n1rs(m),n1re(m)
  530   continue
c$omp end do nowait
c$omp end parallel
c
  540 continue
```

（２）その他の可能性と問題点

外側のループ do 30、do 40、do 540 で並列化できる。試していない。

A12　smcbnd.f

（１）並列化ループ

```
      subroutine smcbnd
c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         iflst,jflst,kflst,imlst,gc,gg,dt,ajac,amach2,rdt,
c$omp&         imach,clp
c$omp&         )
c$omp do
      do 30 n1b=1,n1bmax
   30 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,mvlst,n1dlst,iend,istr,jend,jstr,
c$omp&         iflst,jflst,kflst,xc,yc,zc,x,y,z,amach2,
c$omp&         u,v,w,clp
c$omp&         )
c$omp do
      do 40 n1b=1,n1bmax
   40 continue
c$omp end do nowait
c$omp end parallel

      do 45 n1b=1,n1bmax
   45 continue
```

（２）その他の可能性と問題点

do 45 は小さいので必要なさそうである。

A 13 change1.f

（1） 並列化ループ

```
      subroutine change1(tfluxi,tgco,lcorv)

      do 10 n1b=1,n1b1max
   10 continue

      do 20 n1b=n1b1max+1,n1b2max
   20 continue

      do 30 n1b=n1b2max+1,n1b3max
   30 continue

      do 90 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1,
c$omp&         u,v,w,rho,
c$omp&         uc,vc,wc,gc
c$omp&         )
c$omp do
        do 91 n1=n1rs(m),n1re(m)
   91   continue
c$omp end do nowait
c$omp end parallel
   90 continue

      do 140 m=1,nvol
cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m,n1rs,n1re,iip1,ijp1
cc$omp&         ,n1lst
cc$omp&         ,iflag,jflag,kflag,ipflag
cc$omp&         ,xc,yc,zc
cc$omp&         ,p
cc$omp&         ,dt,ich4
cc$omp&         ,u,v,w
cc$omp&         )
cc$omp do
        do 230 n1=n1rs(m),n1re(m)
  230   continue
cc$omp end do nowait
cc$omp end parallel
c
  140 continue

      do 241 m=1,nvol
  241 continue
      do 141 m=1,nvol
  141 continue

      do 142 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1,
c$omp&         iflag,jflag,kflag,
c$omp&         uc,vc,wc,gc,
c$omp&         u,v,w,rho,
c$omp&         ich4,para,fup,
c$omp&         r6,r16,r56
c$omp&         )
c$omp do
        do 231 n1=n1rs(m),n1re(m)
  231   continue
c$omp end do nowait
c$omp end parallel
c
   142 continue
c
      do 143 m=1,nvol
  143 continue

cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1b1max
cc$omp&         ,iflbnd,n1dbnd,ipbc
cc$omp&         ,uc,ucbnd
cc$omp&         )
cc$omp do
      do 331 n1b=1,n1b1max
  331 continue
cc$omp end do nowait
cc$omp end parallel

cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1b1max,n1b2max
cc$omp&         ,iflbnd,n1dbnd,ipbc
cc$omp&         ,mvbnd,istr,iend
cc$omp&         ,vc,ucbnd
cc$omp&         )
cc$omp do
      do 332 n1b=n1b1max+1,n1b2max
  332 continue
cc$omp end do nowait
cc$omp end parallel

cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1b2max,n1b3max
cc$omp&         ,iflbnd,n1dbnd,ipbc
cc$omp&         ,mvbnd,istr,iend,jstr,jend
cc$omp&         ,wc,ucbnd
cc$omp&         )
cc$omp do
      do 333 n1b=n1b2max+1,n1b3max
  333 continue
cc$omp end do nowait
cc$omp end parallel

cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1b1max
cc$omp&         ,iflbnd,n1dbnd
```

```
cc$omp&         ,mvbnd,itflux,mflux
cc$omp&         ,uc,ucbnd
cc$omp&         ,gc,para
cc$omp&         )
cc$omp do
      do 40 n1b=1,n1b1max
   40 continue
cc$omp end do nowait
cc$omp end parallel


cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1b1max,n1b2max
cc$omp&         ,iflbnd,n1dbnd
cc$omp&         ,mvbnd,istr,iend,itflux,mflux
cc$omp&         ,vc,ucbnd
cc$omp&         ,gc,para
cc$omp&         )
cc$omp do
      do 50 n1b=n1b1max+1,n1b2max
   50 continue
cc$omp end do nowait
cc$omp end parallel


cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1b2max,n1b3max
cc$omp&         ,iflbnd,n1dbnd
cc$omp&         ,mvbnd,istr,iend,jstr,jend,itflux,mflux
cc$omp&         ,wc,ucbnd
cc$omp&         ,gc,para
cc$omp&         )
cc$omp do
      do 60 n1b=n1b2max+1,n1b3max
   60 continue
cc$omp end do nowait
cc$omp end parallel

        do 500 n1=n1rs(1),n1re(nvol)
  500   continue
```

（２）その他の可能性と問題点

コメントアウトしているループも並列化できるが、すべて並列化を行うのがよいかは不明である。比較的小さいのでコメントアウトした。

A 14　change2.f

（1） 並列化ループ

```
      subroutine change2(tfluxi,llast,iwout)
      do 140 m=1,nvol
c
cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m,n1rs,n1re
cc$omp&         ,u,v,w,grad_p,rrho
cc$omp&         )
cc$omp do
        do 230 n1=n1rs(m),n1re(m)
  230 continue
cc$omp end do nowait
cc$omp end parallel
c
  140 continue

      do 141 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1,
c$omp&         gc,gg,n1lst,uc,vc,wc,p,dt
c$omp&         )
c$omp do
        do 231 n1=n1rs(m),n1re(m)
  231 continue
c$omp end do nowait
c$omp end parallel
c
  141 continue


c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,n1dlst,mvlst,
c$omp&         istr,jstr,iend,jend,
c$omp&         iflst,jflst,kflst,imlst,ipbc,
c$omp&         gc,gg,
c$omp&         uc,vc,wc,p,dt
c$omp&         )
c$omp do
      do 241 n1b=1,n1bmax
  241 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,n1dlst,mvlst,
c$omp&         istr,jstr,iend,jend,
c$omp&         iflst,jflst,kflst,imlst,ipbc,
c$omp&         gc,gg,n1lst,
c$omp&         uc,vc,wc,p,dt
c$omp&         )
```

```
c$omp do
      do 242 n1b=1,n1bmax
  242 continue
c$omp end do nowait
c$omp end parallel

      do 142 m=1,nvol
  142 continue


cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1b1max
cc$omp&         ,iflbnd,n1dbnd,ipbc
cc$omp&         ,uc,ucbnd
cc$omp&         )
cc$omp do
      do 331 n1b=1,n1b1max
  331 continue
cc$omp end do nowait
cc$omp end parallel


cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1b1max,n1b2max
cc$omp&         ,iflbnd,n1dbnd,ipbc
cc$omp&         ,mvbnd,istr,iend
cc$omp&         ,vc,ucbnd
cc$omp&         )
cc$omp do
      do 332 n1b=n1b1max+1,n1b2max
  332 continue
cc$omp end do nowait
cc$omp end parallel


cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(n1b2max,n1b3max
cc$omp&         ,iflbnd,n1dbnd,ipbc
cc$omp&         ,mvbnd,istr,iend,jstr,jend
cc$omp&         ,wc,ucbnd
cc$omp&         )
cc$omp do
      do 333 n1b=n1b2max+1,n1b3max
  333 continue
cc$omp end do nowait
cc$omp end parallel

      do 330 n1=n1rs(1),n1re(nvol)
  330 continue
      do 150 m=1,nvol
  150 continue

      do 2005 m=1,nvol
2005 continue
```

（２）その他の可能性と問題点

コメントアウトしているループも並列化できるが、すべて並列化を行うのがよいかは不明である。比較的小さいのでコメントアウトした。

A 15　filter.f

（1） 並列化ループ

```
      subroutine filter(gamma2)
c
      do 140 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1,
c$omp&         ipflag,iflag,jflag,kflag,jqorg,para3,
c$omp&         tu,tv,tw,u,v,w
c$omp&         )
c$omp do
        do 230 n1=n1rs(m),n1re(m)
  230   continue
c$omp end do nowait
c$omp end parallel
c
  140 continue

      do 145 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1,
c$omp&         ipflag,iflag,jflag,kflag,jqorg,para3,
c$omp&         tu,tv,tw,u,v,w
c$omp&         )
c$omp do
        do 235 n1=n1rs(m),n1re(m)
  235 continue
c$omp end do nowait
c$omp end parallel
c
  145 continue

      do 150 m=1,nvol
  150 continue
      do 160 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,nints,ninte,xiin,etin,zein,dlin,
c$omp&         mvlin,istr,iend,jstr,jend,n1in,
c$omp&         tu,tv,tw,uvwbin,n1on
c$omp&         )
c$omp do
        do 550 n=nints(m),ninte(m)
  550 continue
c$omp end do nowait
c$omp end parallel
c
  160 continue
```

（2）その他の可能性と問題点

他の並列化の可能性はない。 do 150 も並列化できるが試していない。

A16　shear.f

（1）　並列化ループ

```
      subroutine shear(isabso)

      do 140 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1,
c$omp&         ipflag,xc,yc,zc,u,v,w,n1lst,
c$omp&         iflst,jflst,kflst,
c$omp&         sij,sabs
c$omp&         )
c$omp do
        do 230 n1=n1rs(m),n1re(m)
  230 continue
c$omp end do nowait
c$omp end parallel
c
  140 continue

      do 150 m=1,nvol
  150 continue

      do 160 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,nints,ninte,xiin,etin,zein,dlin,
c$omp&         mvlin,n1in,istr,iend,jstr,jend,
c$omp&         sij,wbin,n1on,sabs
c$omp&         )
c$omp do
        do 550 n=nints(m),ninte(m)
  550 continue
c$omp end do nowait
c$omp end parallel
  160 continue
c

      do 400 m=1,nvol
cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m,n1rs,n1re,
cc$omp&         sij,clp,itpara,isabso
cc$omp&         )
cc$omp do
        do 410 n1=n1rs(m),n1re(m)
  410   continue
cc$omp end do nowait
cc$omp end parallel
c
  400 continue
```

（２）その他の可能性と問題点

他の並列化の可能性はない。 do 150 も並列化できるが試していない。

A17 dyn.f

（１） 並列化ループ

```
      subroutine dyn (alf_filt,gam2ba0)

      do 10 m=1,nvol
      do 10 n1=n1rs(m),n1re(m)
   10 continue
      do 12 m=1,nvol
      do 12 n1=n1rs(m),n1re(m)
   12 continue
        do 13 m=1,nvol
        do 13 n1=n1rs(m),n1re(m)
   13   continue
      do 14 m=1,nvol
      do 14 n1=n1rs(m),n1re(m)
   14 continue
      do 20 m=1,nvol
      do 20 n1=n1rs(m),n1re(m)
   20 continue
      do 22 m=1,nvol
      do 22 n1=n1rs(m),n1re(m)
   22 continue
      do 30 m=1,nvol
      do 30 n1=n1rs(m),n1re(m)
   30 continue
      do 32 m=1,nvol
      do 32 n1=n1rs(m),n1re(m)
   32 continue
      do 34 m=1,nvol
      do 34 n1=n1rs(m),n1re(m)
   34 continue
      do 50 m=1,nvol
      do 50 n1=n1rs(m),n1re(m)
   50 continue
      do 52 m=1,nvol
      do 52 n1=n1rs(m),n1re(m)
   52 continue
      do 54 m=1,nvol
      do 54 n1=n1rs(m),n1re(m)
   54 continue

      do 140 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,nxm0,nym0,nzm0,ist,iip1,jst,ijp1,kst,n1dstr,
c$omp&         n1rs,n1re,ipflag,iflst,jflst,kflst,
c$omp&         xc,yc,zc,tu,tv,tw,n1lst,
c$omp&         clp,itpara,iflag,jflag,kflag,sij
c$omp&         )
c$omp do
        do 230 n1=n1rs(m),n1re(m)
  230 continue
c$omp end do nowait
c$omp end parallel
c
  140 continue
```

```
      do 200 m=1,nvol

          do 319 j=1,nym0-1
  319     continue
        do 310 k=1,nzm0-1
          do 330 i=1,nxm0-1
            do 320 j=1,nym0-1
  320       continue
  330     continue
  310   continue
        do 311 k=1,nzm0-1
          do 321 j=1,nym0-1
            do 331 i=1,nxm0-1
  331       continue
  321     continue
  311   continue
c
  200 continue

      do 250 m=1,nvol
        do 369 j=1,nym0-1
          do 379 i=1,nxm0-1
  379     continue
  369   continue
        do 360 k=1,nzm0-1
          do 370 j=1,nym0-1
            do 380 i=1,nxm0-1
  380       continue
  370     continue
  360   continue
          do 371 j=1,nym0-1
            do 381 i=1,nxm0-1
  381       continue
  371     continue
  361   continue
  250 continue

      do 201 m=1,nvol
cc$omp parallel
cc$omp&  default(private)
cc$omp&  shared(m,n1rs,n1re,
cc$omp&         clp,iba,idyrv,ajac,r3
cc$omp&         )
cc$omp do
        do 210 n1=n1rs(m),n1re(m)
  210   continue
cc$omp end do nowait
cc$omp end parallel
  201 continue

      do 145 m=1,1
        do 420 jj=1,(nym0-1)/2
        do 421 j=jj,nym0-jj,nym0-2*jj
  422       continue
  410     continue
  421   continue
  420   continue
  145 continue
```

```
    do 146 m=1,1
      do 430 jj=1,(nym0-1)/2
      do 441 j=jj,nym0-jj,nym0-2*jj
        do 440 k=1,nzm0-1
          do 442 i=1,nxm0-1
442       continue
440     continue
441   continue
430 continue
146 continue
    do 144 ik=1,ikind
144 continue
    do 150 m=1,nvol
150 continue
```

（２）その他の可能性と問題点

do140 および do201 のほかは変数の置き換えなので並列化の必要はなさそうである。do201 も小さいのでコメントアウトした。

A18　sgscal.f

（1） 並列化ループ

```
      subroutine sgscal

      do 10 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1,
c$omp&         ipflag,
c$omp&         xc,yc,zc,tk,
c$omp&         itflag,tkes,rddes_d,dles,damp,
c$omp&         amu,r6,rrho,epstk,tesgs,yeps
c$omp&         )
c$omp do
       do 100 n1=n1rs(m),n1re(m)
  100  continue
c$omp end do nowait
c$omp end parallel
c
   10 continue


c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,n1dlst,mvlst,n1bnd,iflst,jflst,kflst,
c$omp&         istr,iend,jstr,jend,
c$omp&         x,y,z,tkbnd,
c$omp&         tk,itflag,tkes,rddes_d,dles,damp,
c$omp&         amu,r6,rrho,epstk,tesgs,yeps
c$omp&         )
c$omp do
      do 20 n1b=1,n1bmax
   20 continue
c$omp end do nowait
c$omp end parallel



      do 800 m=1,nvol
        do 422 k=0,nzm0
          do 424 j=0,nym0
  424     continue
  422   continue
        do 432 k=0,nzm0
          do 434 i=ist,ist+iip1-1
  434     continue
  432   continue
        do 442 n1k0=n1st,n1st+ijp1-1
  442   continue
        do 607 k=0,nzm0
          do 606 j=0,nym0
  606     continue
  607   continue
        do 609 k=0,nzm0
          do 608 j=0,nym0
  608     continue
  609   continue
```

```
       do 603 k=0,nzm0
         do 602 i=ist,ist+iip1-1
602      continue
603    continue
      do 605 k=0,nzm0
         do 604 i=ist,ist+iip1-1
604      continue
605    continue
       do 600 n1k0=n1st,n1st+ijp1-1
600    continue
       do 601 n1nz=n1st,n1st+ijp1-1
601    continue
       do 707 k=0,nzm0
         do 706 j=0,nym0
706      continue
707    continue
       do 709 k=0,nzm0
         do 708 j=0,nym0
708      continue
709    continue
       do 703 k=0,nzm0
         do 702 i=ist,ist+iip1-1
702      continue
703    continue
       do 705 k=0,nzm0
         do 704 i=ist,ist+iip1-1
704      continue
705    continue
       do 710 i=0,nxm0
         do 700 j=0,nym0
700      continue
710    continue
       do 711 i=0,nxm0
         do 701 j=0,nym0
701      continue
711    continue

800 continue
```

（２）その他の可能性と問題点

do800 は変数の置き換えなので並列化は行わなかった。

A19　calc_grad_p.f

（1）　並列化ループ

```
      subroutine calc_grad_p

      do 140 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,iip1,ijp1,
c$omp&         n1lst,ipflag,
c$omp&         xc,yc,zc,dt,ich4,p,
c$omp&         iflag,jflag,kflag,grad_p
c$omp&         )
c$omp do
        do 240 n1=n1rs(m),n1re(m)
  240 continue
c$omp end do nowait
c$omp end parallel
c
  140 continue


c$omp parallel
c$omp&  default(private)
c$omp&  shared(n1bmax,n1dlst,mvlst,
c$omp&         istr,iend,jstr,jend,
c$omp&         xc,yc,zc,p,x,y,z,iflst,jflst,kflst,
c$omp&         ipbc,dt,iflag,jflag,kflag,ich4,grad_p
c$omp&         )
c$omp do
      do 150 n1b=1,n1bmax
  150 continue
c$omp end do nowait
c$omp end parallel
```

（2）その他の可能性と問題点

他の並列化の可能性はない。

A20　intp_2.f

（１）　並列化ループ

```
      subroutine intp(m)

c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,difsqr,sumsqr,nints,ninte,
c$omp&         xiin,etin,zein,mvlin,
c$omp&         istr,iend,jstr,jend,nlin,
c$omp&         p,nlon,bp
c$omp&         )
c$omp do reduction(+ : difsqr,sumsqr)
      do 200 n=nints(m),ninte(m)
  200 continue
c$omp end do nowait
c$omp end parallel
```

（２）その他の可能性と問題点

他の並列化の可能性はない。

A21　intv_2.f

（1）　並列化ループ

```
      subroutine intv(m)

c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,nints,ninte,
c$omp&         xiin,etin,zein,dlin,mvlin,
c$omp&         istr,iend,jstr,jend,nlin,
c$omp&         u,v,w,uvwbin,nlon,bu,bv,bw
c$omp&         )
c$omp do
      do 200 n=nints(m),ninte(m)
  200 continue
c$omp end do nowait
c$omp end parallel
```

（2）その他の可能性と問題点

他の並列化の可能性はない。

A22 wallf0.f

（１） 並列化ループ

```
      subroutine wallf0

c$omp parallel
c$omp&  default(private)
c$omp&  shared(a1,b1,a2,b2,paralmt,tkappa,rtkappa,
c$omp&         pr1,bt,ondo_yplslmt,rre,rreb1,rreb2,
c$omp&         yplsc1,yplsc2,bb1,ab1,bb2,ab2,r2b1,r2b2,
c$omp&         fa1,fb1,fa2,fb2,fc2,ga1,gb1,ga2,ymasc1,ymasc2,
c$omp&         lc1,lc2,lc3,
c$omp&         mvwall,n1wall,n1lst,
c$omp&         iflst,jflst,kflst,n1bnd,
c$omp&         u,v,w,uvwbnd,dd1,re,rhobnd,
c$omp&         cx1,cy1,cz1,dzero,parbnd,amu_t,amu,
c$omp&         iwbc,idp2,ihmtfl,te,tebnd,alf_t,alf
c$omp&         )
c$omp do
      do 700 lc=1,lc1
  700 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(a1,b1,a2,b2,paralmt,tkappa,rtkappa,
c$omp&         pr1,bt,ondo_yplslmt,rre,rreb1,rreb2,
c$omp&         yplsc1,yplsc2,bb1,ab1,bb2,ab2,r2b1,r2b2,
c$omp&         fa1,fb1,fa2,fb2,fc2,ga1,gb1,ga2,ymasc1,ymasc2,
c$omp&         lc1,lc2,lc3,
c$omp&         mvwall,n1wall,n1lst,
c$omp&         iflst,jflst,kflst,n1bnd,
c$omp&         u,v,w,uvwbnd,dd1,re,rhobnd,
c$omp&         cx1,cy1,cz1,dzero,parbnd,amu_t,amu,
c$omp&         iwbc,idp2,ihmtfl,te,tebnd,alf_t,alf
c$omp&         )
c$omp do
      do 720 lc=lc1+1,lc2
  720 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(a1,b1,a2,b2,paralmt,tkappa,rtkappa,
c$omp&         pr1,bt,ondo_yplslmt,rre,rreb1,rreb2,
c$omp&         yplsc1,yplsc2,bb1,ab1,bb2,ab2,r2b1,r2b2,
c$omp&         fa1,fb1,fa2,fb2,fc2,ga1,gb1,ga2,ymasc1,ymasc2,
c$omp&         lc1,lc2,lc3,
c$omp&         mvwall,n1wall,n1lst,
c$omp&         iflst,jflst,kflst,n1bnd,
c$omp&         u,v,w,uvwbnd,dd1,re,rhobnd,
c$omp&         cx1,cy1,cz1,dzero,parbnd,amu_t,amu,
c$omp&         iwbc,idp2,ihmtfl,te,tebnd,alf_t,alf
c$omp&         )
```

```
c$omp do
      do 740 lc=lc2+1,lc3
  740 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(a1,b1,a2,b2,paralmt,tkappa,rtkappa,
c$omp&         pr1,bt,ondo_yplslmt,rre,rreb1,rreb2,
c$omp&         yplsc1,yplsc2,bb1,ab1,bb2,ab2,r2b1,r2b2,
c$omp&         fa1,fb1,fa2,fb2,fc2,ga1,gb1,ga2,ymasc1,ymasc2,
c$omp&         lcg1,lcg2,lcg3,mvwalg,n1walg,n1lst,
c$omp&         iflst,jflst,kflst,n1bnd,
c$omp&         u,v,w,uvwbnd,dd1g,re,rhobnd,
c$omp&         gc,dzero,parbnd,amu_t,amu,
c$omp&         nslip,idp2,ihmtfl,te,tebnd,alf_t,alf
c$omp&         )
c$omp do
      do 800 lc=1,lcg1
  800 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(a1,b1,a2,b2,paralmt,tkappa,rtkappa,
c$omp&         pr1,bt,ondo_yplslmt,rre,rreb1,rreb2,
c$omp&         yplsc1,yplsc2,bb1,ab1,bb2,ab2,r2b1,r2b2,
c$omp&         fa1,fb1,fa2,fb2,fc2,ga1,gb1,ga2,ymasc1,ymasc2,
c$omp&         lcg1,lcg2,lcg3,mvwalg,n1walg,n1lst,
c$omp&         iflst,jflst,kflst,n1bnd,
c$omp&         u,v,w,uvwbnd,dd1g,re,rhobnd,
c$omp&         gc,dzero,parbnd,amu_t,amu,
c$omp&         nslip,idp2,ihmtfl,te,tebnd,alf_t,alf
c$omp&         )
c$omp do
      do 820 lc=lcg1+1,lcg2
  820 continue
c$omp end do nowait
c$omp end parallel

c$omp parallel
c$omp&  default(private)
c$omp&  shared(a1,b1,a2,b2,paralmt,tkappa,rtkappa,
c$omp&         pr1,bt,ondo_yplslmt,rre,rreb1,rreb2,
c$omp&         yplsc1,yplsc2,bb1,ab1,bb2,ab2,r2b1,r2b2,
c$omp&         fa1,fb1,fa2,fb2,fc2,ga1,gb1,ga2,ymasc1,ymasc2,
c$omp&         lcg1,lcg2,lcg3,mvwalg,n1walg,n1lst,
c$omp&         iflst,jflst,kflst,n1bnd,
c$omp&         u,v,w,uvwbnd,dd1g,re,rhobnd,
c$omp&         gc,dzero,parbnd,amu_t,amu,
c$omp&         nslip,idp2,ihmtfl,te,tebnd,alf_t,alf
c$omp&         )
c$omp do
      do 840 lc=lcg2+1,lcg3
      do 840 lc=lcg2+1,lcg3
  840 continue
c$omp end do nowait
c$omp end parallel
```

（２）その他の可能性と問題点

他の並列化の可能性はない。

A23　lawall.f

（１）　並列化ループ

```
      subroutine lawall

      do 140 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,
c$omp&         iflag,jflag,kflag,
c$omp&         clp
c$omp&         )
c$omp do
        do 141 n1=n1rs(m),n1re(m)
  141   continue
c$omp end do nowait
c$omp end parallel
  140 continue
```

（２）その他の可能性と問題点

他の並列化の可能性はない。

A24　cdsl.f

（１）　並列化ループ

```
      subroutine cdsl(rasum4,kt)
c$omp parallel
c$omp&  default(private)
c$omp&  shared(fdp1,flp1,fsp1,pref,
c$omp&         lc1,lc2,lc3,mvwall,nlwall,
c$omp&         amu_t,amu,cx1,cy1,cz1,rwall,
c$omp&         u,v,w,dd1,icdall,p
c$omp&         )
c$omp do reduction(+ : fdp1,flp1,fsp1 )
      do 700 lc=1,lc1
c*aas->変数書き換え
          tmpx = -cx1(lc)*rwall(lc)*pm31
          tmpy = -cy1(lc)*rwall(lc)*pm31
          tmpz = -cz1(lc)*rwall(lc)*pm31

        fdp1=fdp1 + tmpx
        flp1=flp1 + tmpy
        fsp1=fsp1 + tmpz

        fdp1=fdp1 + us*rrys*rwall(lc)
        flp1=flp1 + vs*rrys*rwall(lc)
        fsp1=fsp1 + ws*rrys*rwall(lc)
c*aas        fdp1=-cx1(lc)*rwall(lc)*pm31+fdp1
c*aas        flp1=-cy1(lc)*rwall(lc)*pm31+flp1
c*aas        fsp1=-cz1(lc)*rwall(lc)*pm31+fsp1
c*aas        fdp1=us*rrys*rwall(lc)+fdp1
c*aas        flp1=vs*rrys*rwall(lc)+flp1
c*aas        fsp1=ws*rrys*rwall(lc)+fsp1
c*aas<-
  700 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(fdp1,flp1,fsp1,pref,
c$omp&         lc1,lc2,lc3,mvwall,nlwall,
c$omp&         istr,iend,jstr,jend,
c$omp&         amu_t,amu,cx1,cy1,cz1,
c$omp&         u,v,w,dd1,icdall,p,rwall
c$omp&         )
c$omp do reduction(+ : fdp1,flp1,fsp1 )
      do 720 lc=lc1+1,lc2
c*aas->
          tmpx = -cx1(lc)*rwall(lc)*pm31
          tmpy = -cy1(lc)*rwall(lc)*pm31
          tmpz = -cz1(lc)*rwall(lc)*pm31

        fdp1=fdp1 + tmpx
        flp1=flp1 + tmpy
        fsp1=fsp1 + tmpz

        fdp1=fdp1 + us*rrys*rwall(lc)
        flp1=flp1 + vs*rrys*rwall(lc)
        fsp1=fsp1 + ws*rrys*rwall(lc)
c*aas        fdp1=-cx1(lc)*rwall(lc)*pm31+fdp1
c*aas        flp1=-cy1(lc)*rwall(lc)*pm31+flp1
c*aas        fsp1=-cz1(lc)*rwall(lc)*pm31+fsp1
c*aas        fdp1=us*rrys*rwall(lc)+fdp1
c*aas        flp1=vs*rrys*rwall(lc)+flp1
c*aas        fsp1=ws*rrys*rwall(lc)+fsp1
c*aas<-
  720 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(fdp1,flp1,fsp1,pref,
c$omp&         lc1,lc2,lc3,mvwall,nlwall,
c$omp&         istr,iend,jstr,jend,
c$omp&         amu_t,amu,cx1,cy1,cz1,
c$omp&         u,v,w,dd1,icdall,p,rwall,
c$omp&         cdrag,clift,cside
c$omp&         )
c$omp do reduction(+ : fdp1,flp1,fsp1 )
      do 740 lc=lc2+1,lc3
c*aas->
          tmpx = -cx1(lc)*rwall(lc)*pm31
          tmpy = -cy1(lc)*rwall(lc)*pm31
          tmpz = -cz1(lc)*rwall(lc)*pm31

        fdp1=fdp1 + tmpx
        flp1=flp1 + tmpy
        fsp1=fsp1 + tmpz

        fdp1=fdp1 + us*rrys*rwall(lc)
        flp1=flp1 + vs*rrys*rwall(lc)
        fsp1=fsp1 + ws*rrys*rwall(lc)
c*aas        fdp1=-cx1(lc)*rwall(lc)*pm31+fdp1
c*aas        flp1=-cy1(lc)*rwall(lc)*pm31+flp1
c*aas        fsp1=-cz1(lc)*rwall(lc)*pm31+fsp1
c*aas        fdp1=us*rrys*rwall(lc)+fdp1
c*aas        flp1=vs*rrys*rwall(lc)+flp1
c*aas        fsp1=ws*rrys*rwall(lc)+fsp1
c*aas<-
  740 continue
c$omp end do nowait
c$omp end parallel
```

（２）その他の可能性と問題点

他の並列化の可能性はない。
do reduction(+ : fdp1,flp1,fsp1)を使うため、変数の書き換えを行った。

A25　favr19.f

（1） 並列化ループ

```
      subroutine favr19 (kt)

            do 140 m=1,nvol
            do 140 n1=n1rs(m),n1re(m)
  140       continue

            do 130 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(ktavrw,tdif,tcef,tden,
c$omp&          m,n1rs,n1re,
c$omp&          dt,favr,p,u,v,w,te,anu_t,alf_t,
c$omp&          work,cdyn,tdyn,ajac,idfsgs
c$omp&          )
c$omp do
            do 131 n1=n1rs(m),n1re(m)  !*aas
  131       continue   !*aas
c$omp end do nowait
c$omp end parallel

  130       continue
```

（2）その他の可能性と問題点

do 140 は初期化だけなので必要ない。他の並列化の可能性はない。

A26　ucdiff.f

（１） 並列化ループ

```
      subroutine ucdiff

      do 340 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,
c$omp&         gc,u,v,w,uc,vc,wc,rho,
c$omp&         pcup,ipflag,cup4v
c$omp&         )
c$omp do
        do 330 n1=n1rs(m),n1re(m)
  330   continue
c$omp end do nowait
c$omp end parallel


        do 522 k=0,nzm0
  524     continue
  522   continue
        do 532 k=0,nzm0
          do 534 i=ist,ist+iip1-1
  534     continue
  532   continue
        do 542 n1k0=n1st,n1st+ijp1-1
  542   continue
        do 707 k=0,nzm0
          do 706 j=0,nym0
  706     continue
  707   continue
        do 709 k=0,nzm0
          do 708 j=0,nym0
  708     continue
  709   continue
        do 703 k=0,nzm0
          do 702 i=ist,ist+iip1-1
  702     continue
  703   continue
        do 705 k=0,nzm0
          do 704 i=ist,ist+iip1-1
  704     continue
  705   continue
        do 700 n1k0=n1st,n1st+ijp1-1
  700   continue
        do 701 n1nz=n1st,n1st+ijp1-1
  701   continue
  340 continue

     do 800 m=1,nvol
        do 810 n1=n1rs(m),n1re(m)
  810   continue
        do 815 n1=n1rs(m),n1re(m)
  815   continue
        do 820 n1=n1rs(m),n1re(m)
  820   continue
```

```
        do 825 n1=n1rs(m),n1re(m)
  825   continue
        do 830 n1=n1rs(m),n1re(m)
  830   continue
        do 835 n1=n1rs(m),n1re(m)
  835   continue
  800 continue

      do 140 m=1,nvol
c$omp parallel
c$omp&  default(private)
c$omp&  shared(m,n1rs,n1re,
c$omp&         uc,vc,wc,div,cup4v,
c$omp&         ipflag,ipcup
c$omp&         )
c$omp do
        do 230 n1=n1rs(m),n1re(m)
  230   continue
c$omp end do nowait
c$omp end parallel


        do 422 k=0,nzm0
          do 424 j=0,nym0
  424     continue
  422   continue
        do 432 k=0,nzm0
          do 434 i=ist,ist+iip1-1
  434     continue
  432   continue
        do 442 n1k0=n1st,n1st+ijp1-1
  442   continue
        do 607 k=0,nzm0
          do 606 j=0,nym0
  606     continue
  607   continue
        do 609 k=0,nzm0
          do 608 j=0,nym0
  608     continue
  609   continue
        do 603 k=0,nzm0
          do 602 i=ist,ist+iip1-1
  602     continue
  603   continue
        do 605 k=0,nzm0
          do 604 i=ist,ist+iip1-1
  604     continue
  605   continue
        do 600 n1k0=n1st,n1st+ijp1-1
  600   continue
        do 601 n1nz=n1st,n1st+ijp1-1
  601   continue
c
  140 continue

      do 900 m=1,nvol
        do 915 n1=n1rs(m),n1re(m)
  915   continue
```

```
        do 572 k=0,nzm0
          do 574 j=0,nym0
  574     continue
  572   continue
        do 582 k=0,nzm0
          do 584 i=ist,ist+iipl-1
  584     continue
  582   continue
        do 592 n1k0=n1st,n1st+ijpl-1
  592   continue
        do 757 k=0,nzm0
          do 756 j=0,nym0
  756     continue
  757   continue
          do 758 j=0,nym0
  758     continue
  759   continue
        do 753 k=0,nzm0
          do 752 i=ist,ist+iipl-1
  752     continue
  753   continue
        do 755 k=0,nzm0
          do 754 i=ist,ist+iipl-1
  754     continue
  755   continue
        do 750 n1k0=n1st,n1st+ijpl-1
  750   continue
        do 751 n1nz=n1st,n1st+ijpl-1
  751   continue
  900 continue

      do 1800 m=1,nvol
        do 1810 nl=nlrs(m),nlre(m)
1810   continue
        do 1815 nl=nlrs(m),nlre(m)
1815   continue
        do 1820 nl=nlrs(m),nlre(m)
1820   continue
        do 1825 nl=nlrs(m),nlre(m)
1825   continue
        do 1830 nl=nlrs(m),nlre(m)
1830   continue
        do 1835 nl=nlrs(m),nlre(m)
1835   continue
 1800 continue
```

（２）その他の可能性と問題点

外側のループ do 340、 do 140 で行うこともできるが試していない。その他のループは並列化のメリットはなさそうである。

A27 dampfm.f

（1） 並列化ループ

```
      subroutine dampf

      do 140 m=1,nvol
        do 200 nl=nlrs(m),nlre(m)
  200   continue
  140 continue

c$omp parallel
c$omp&  default(private)
c$omp&  shared(lc1,lc2,lc3,
c$omp&         mvwall,istr,iend,jstr,jend,nlwall,
c$omp&         x,y,z,irdamp,nlrs,nlre,
c$omp&         xc,yc,zc,damp,nvol
c$omp&         )
c$omp do
      do 310 lc=1,lc1
          do 210 nl=nlrs(mw),nlre(mw)
  210     continue
          do 211 m=1,nvol
          do 211 nl=nlrs(m),nlre(m)
  211     continue
  310 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(lc1,lc2,lc3,
c$omp&         mvwall,istr,iend,jstr,jend,nlwall,
c$omp&         x,y,z,irdamp,nlrs,nlre,
c$omp&         xc,yc,zc,damp,nvol
c$omp&         )
c$omp do
      do 320 lc=lc1+1,lc2
c
          do 220 nl=nlrs(mw),nlre(mw)
  220     continue
          do 221 m=1,nvol
          do 221 nl=nlrs(m),nlre(m)
  221     continue
  320 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(lc1,lc2,lc3,
c$omp&         mvwall,istr,iend,jstr,jend,nlwall,
c$omp&         x,y,z,irdamp,nlrs,nlre,
c$omp&         xc,yc,zc,damp,nvol
c$omp&         )
c$omp do
```

```
      do 330 lc=lc2+1,lc3
          do 230 n1=n1rs(mw),n1re(mw)
  230     continue
          do 231 m=1,nvol
          do 231 n1=n1rs(m),n1re(m)
  231     continue
  330 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(lcg1,lcg2,lcg3,
c$omp&         mvwall,istr,iend,jstr,jend,n1wall,
c$omp&         x,y,z,irdamp,n1rs,n1re,
c$omp&         xc,yc,zc,damp,nvol
c$omp&         )
c$omp do
      do 410 lcg=1,lcg1
c
          do 240 n1=n1rs(mw),n1re(mw)
  240     continue
          do 241 m=1,nvol
          do 241 n1=n1rs(m),n1re(m)
  241     continue
  410 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(lcg1,lcg2,lcg3,
c$omp&         mvwall,istr,iend,jstr,jend,n1wall,
c$omp&         x,y,z,irdamp,n1rs,n1re,
c$omp&         xc,yc,zc,damp,nvol
c$omp&         )
c$omp do
      do 420 lcg=lcg1+1,lcg2
          do 250 n1=n1rs(mw),n1re(mw)
  250     continue
          do 251 m=1,nvol
          do 251 n1=n1rs(m),n1re(m)
  251     continue
  420 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(lcg1,lcg2,lcg3,
c$omp&         mvwall,istr,iend,jstr,jend,n1wall,
c$omp&         x,y,z,irdamp,n1rs,n1re,
c$omp&         xc,yc,zc,damp,nvol
c$omp&         )
c$omp do
```

```
      do 430 lcg=lcg2+1,lcg3
c
          do 260 n1=n1rs(mw),n1re(mw)
  260     continue
          do 261 m=1,nvol
          do 261 n1=n1rs(m),n1re(m)
  261     continue
  430 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(nvol,nx,ny,nz,
c$omp&         istr,iend,jstr,jend,kstr,n1ds,
c$omp&         itype,jtype,ktype,damp,n1dstr
c$omp&         )
c$omp do
      do 150 m=1,nvol
        do 422 k=0,nzm0
          do 424 j=0,nym0
  424     continue
  422   continue
        do 432 k=0,nzm0
          do 434 i=ist,ist+iip1-1
  434     continue
  432   continue
        do 442 n1k0=n1st,n1st+ijp1-1
  442   continue
      endif
        do 607 k=0,nzm0
          do 606 j=0,nym0
  606     continue
  607   continue
        do 609 k=0,nzm0
          do 608 j=0,nym0
  608     continue
  609   continue
          do 702 j=0,nym0
  702     continue
  701   continue
      endif
        do 603 k=0,nzm0
          do 602 i=ist,ist+iip1-1
  602     continue
  603   continue
        do 605 k=0,nzm0
          do 604 i=ist,ist+iip1-1
  604     continue
  605   continue
        do 703 k=0,nzm0
          do 704 i=ist,ist+iip1-1
  704     continue
  703   continue
        do 600 n1k0=n1st,n1st+ijp1-1
  600   continue
        do 601 n1nz=n1st,n1st+ijp1-1
  601   continue
        do 705 i=0,nxm0
          do 706 j=0,nym0
  706     continue
  705   continue
  150 continue
c$omp end do nowait
c$omp end parallel


c$omp parallel
c$omp&  default(private)
c$omp&  shared(nvol,nints,ninte,xiin,etin,zein,dlin,
c$omp&         mvlin,istr,iend,jstr,jend,kstr,n1in,n1on,
c$omp&         damp,dmbin,nx,ny,nz,n1ds,
c$omp&         itype,jtype,ktype,n1dstr
c$omp&         )
c$omp do
      do 160 m=1,nvol
        do 550 n=nints(m),ninte(m)
  550 continue
c%%% for cyclic condition %%%
        do 522 k=0,nzm0
          do 524 j=0,nym0
  524     continue
  522   continue
        do 532 k=0,nzm0
          do 534 i=ist,ist+iip1-1
  534     continue
  532   continue
        do 542 n1k0=n1st,n1st+ijp1-1
  542   continue
        do 507 k=0,nzm0
          do 506 j=0,nym0
  506     continue
  507   continue
      endif
        do 509 k=0,nzm0
          do 508 j=0,nym0
  508     continue
  509   continue
        do 711 k=0,nzm0
          do 712 j=0,nym0
  712     continue
  711   continue
        do 503 k=0,nzm0
          do 502 i=ist,ist+iip1-1
  502     continue
  503   continue
        do 505 k=0,nzm0
          do 504 i=ist,ist+iip1-1
  504     continue
  505   continue
        do 713 k=0,nzm0
          do 714 i=ist,ist+iip1-1
  714     continue
  713   continue
        do 500 n1k0=n1st,n1st+ijp1-1
  500   continue
        do 501 n1nz=n1st,n1st+ijp1-1
  501   continue
        do 715 i=0,nxm0
          do 716 j=0,nym0
  716     continue
  715   continue
  160 continue
c$omp end do nowait
c$omp end parallel
```

（２）その他の可能性と問題点

内側のループが小さいので、外側のループを並列化した。