Add explicit group vector to acc, needed by CRAY compilers #298

Edit Code

Merged **sfiligoi** merged 3 commits into `gafusion:master` from `sfiligoi:crusher_230227` on Feb 27

Conversation 0 Commits 3 Checks 0 Files changed 9 +67 −35

Changes from **all commits** File filter Conversations

Filter changed files

- cgyro/src
  - cgyro_error_estimate.F90
  - cgyro_field.F90
  - cgyro_math.F90
  - cgyro_nl_comm.F90
  - cgyro_nl_fftw.gpu.F90
  - cgyro_parallel_lib.F90
  - cgyro_rhs.gpu.f90
  - cgyro_source.F90
  - cgyro_upwind.F90

3 cgyro/src/cgyro_error_estimate.F90

```
29    ! NOTE: If I have multiple itor, sum them all together
30    h_s=0.0
31    r_s=0.0
32  - !$acc parallel loop collapse(3) independent present(h_x,rhs(:,:,:,1)) reduction(+:h_s,r_s) async(2)
33    do itor=nt1,nt2
34     do iv_loc=1,nv_loc
35       do ic=1,nc
```

```
29    ! NOTE: If I have multiple itor, sum them all together
30    h_s=0.0
31    r_s=0.0
32  + !$acc parallel loop collapse(3) independent gang vector &
33  + !$acc&         present(h_x,rhs(:,:,:,1)) reduction(+:h_s,r_s) async(2)
34    do itor=nt1,nt2
35     do iv_loc=1,nv_loc
36       do ic=1,nc
```

24 cgyro/src/cgyro_field.F90

```
169
170   ! Poisson and Ampere RHS integrals of H
171
172 - !$acc parallel loop collapse(3) independent copyin(start_t) &
173   !$acc&         present(nt2,nc,n_field) default(none)
174    do itor=start_t,nt2
175     do ic=1,nc
224
225   call timer_lib_in('field')
226   ! Poisson LHS factors
227 - !$acc parallel loop collapse(3) independent present(fcoef) copyin(start_t) &
228   !$acc&         present(nt2,nc,n_field) default(none)
229    do itor=start_t,nt2
230       ! assuming  (.not.(itor == 0 .and. ae_flag == 1))
443
444   ! Poisson and Ampere RHS integrals of h
445
446 - !$acc parallel loop collapse(3) independent private(field_loc_l) &
447   !$acc&         present(dvjvec_c) present(nt1,nt2,nc,n_field,nv1,nv2) default(none)
448    do itor=nt1,nt2
449     do ic=1,nc
484   call timer_lib_out('field_com')
485   call timer_lib_in('field')
486   if (n_field > 2) then
487 - !$acc parallel loop collapse(2) independent present(fcoef) &
488   !$acc&         present(nt1,nt2,nc) default(none)
489      do itor=nt1,nt2
490       do ic=1,nc
508
509   if (itor1<=itor2) then
510     if (n_field > 2) then
511 - !$acc parallel loop collapse(2) independent private(tmp) present(gcoef) &
512   !$acc&         copyin(itor1,itor2) present(nc) default(none)
513         do itor=itor1,itor2
514          do ic=1,nc
521          enddo
522         enddo
523       else
524 - !$acc parallel loop collapse(3) independent present(gcoef) &
525   !$acc&         copyin(itor1,itor2) present(nc,n_field) default(none)
526         do itor=itor1,itor2
527          do ic=1,nc
570
571   ! Poisson and Ampere RHS integrals of h
572
573 - !$acc parallel loop collapse(3) independent private(field_loc_l) &
574   !$acc&         present(dvjvec_c) present(nc,n_field,nv1,nv2) default(none)
575    do itor=0,0
576     do ic=1,nc
611   call timer_lib_out('field_com')
612   call timer_lib_in('field')
613   if (n_field > 2) then
614 - !$acc parallel loop collapse(2) independent present(fcoef) present(nc) default(none)
615      do itor=0,0
616        do ic=1,nc
617         field(3,ic,itor) = field(3,ic,itor)*fcoef(3,ic,itor)
```

```
169
170   ! Poisson and Ampere RHS integrals of H
171
172 + !$acc parallel loop collapse(3) gang vector &
173 + !$acc&         independent copyin(start_t) &
174   !$acc&         present(nt2,nc,n_field) default(none)
175    do itor=start_t,nt2
176     do ic=1,nc
225
226   call timer_lib_in('field')
227   ! Poisson LHS factors
228 + !$acc parallel loop collapse(3) gang vector &
229 + !$acc&         independent present(fcoef) copyin(start_t) &
230   !$acc&         present(nt2,nc,n_field) default(none)
231    do itor=start_t,nt2
232       ! assuming  (.not.(itor == 0 .and. ae_flag == 1))
445
446   ! Poisson and Ampere RHS integrals of h
447
448 + !$acc parallel loop collapse(3) gang vector &
449 + !$acc&         independent private(field_loc_l) &
450   !$acc&         present(dvjvec_c) present(nt1,nt2,nc,n_field,nv1,nv2) default(none)
451    do itor=nt1,nt2
452     do ic=1,nc
487   call timer_lib_out('field_com')
488   call timer_lib_in('field')
489   if (n_field > 2) then
490 + !$acc parallel loop collapse(2) gang vector &
491 + !$acc&         independent present(fcoef) &
492   !$acc&         present(nt1,nt2,nc) default(none)
493      do itor=nt1,nt2
494       do ic=1,nc
512
513   if (itor1<=itor2) then
514     if (n_field > 2) then
515 + !$acc parallel loop collapse(2) gang vector &
516 + !$acc&         independent private(tmp) present(gcoef) &
517   !$acc&         copyin(itor1,itor2) present(nc) default(none)
518         do itor=itor1,itor2
519          do ic=1,nc
526          enddo
527         enddo
528       else
529 + !$acc parallel loop collapse(3) gang vector &
530 + !$acc&         independent present(gcoef) &
531   !$acc&         copyin(itor1,itor2) present(nc,n_field) default(none)
532         do itor=itor1,itor2
533          do ic=1,nc
576
577   ! Poisson and Ampere RHS integrals of h
578
579 + !$acc parallel loop collapse(3) gang vector &
580 + !$acc&         independent private(field_loc_l) &
581   !$acc&         present(dvjvec_c) present(nc,n_field,nv1,nv2) default(none)
582    do itor=0,0
583     do ic=1,nc
618   call timer_lib_out('field_com')
619   call timer_lib_in('field')
620   if (n_field > 2) then
621 + !$acc parallel loop collapse(2) gang vector &
622 + !$acc&         independent present(fcoef) present(nc) default(none)
623      do itor=0,0
624        do ic=1,nc
625         field(3,ic,itor) = field(3,ic,itor)*fcoef(3,ic,itor)
```

46 cgyro/src/cgyro_math.F90

```
26      integer :: i
27      !---------------------------------------------------
28    #ifdef _OPENACC
29  - !$acc parallel loop independent present(left,r1)
30      do i=1,sz
31         left(i) = r1(i)
32      enddo
51      complex :: tmp
52      !---------------------------------------------------
53    #ifdef _OPENACC
54  - !$acc parallel loop independent present(left1,left2,r1) private(tmp)
55      do i=1,sz
56         tmp = r1(i)
57         left1(i) = tmp
89      if (present(abssum)) then
90        s = 0.0
91    #ifdef _OPENACC
92  - !$acc parallel loop independent present(left,r1,r2) private(tmp) reduction(+:s)
93    #else
94    !$omp parallel do private(tmp) reduction(+:s)
95    #endif
101        abssum = s
102     else
103   #ifdef _OPENACC
104 - !$acc parallel loop independent present(left,r1,r2)
105   #else
106   !$omp parallel do
107   #endif
131     if (present(abssum)) then
132       s = 0.0
133   #ifdef _OPENACC
134 - !$acc parallel loop independent present(left,r1,r2,r3) private(tmp) reduction(+:s)
135   #else
136   !$omp parallel do private(tmp) reduction(+:s)
137   #endif
143        abssum = s
144     else
145   #ifdef _OPENACC
146 - !$acc parallel loop independent present(left,r1,r2,r3)
147   #else
148   !$omp parallel do
149   #endif
175     if (present(abssum)) then
176       s = 0.0
177   #ifdef _OPENACC
178 - !$acc parallel loop independent present(left,r1,r2,r3,r4) private(tmp) reduction(+:s)
179   #else
180   !$omp parallel do private(tmp) reduction(+:s)
181   #endif
187        abssum = s
188     else
189   #ifdef _OPENACC
190 - !$acc parallel loop independent present(left,r1,r2,r3,r4)
191   #else
192   !$omp parallel do
193   #endif
221     if (present(abssum)) then
222       s = 0.0
223   #ifdef _OPENACC
224 - !$acc parallel loop independent present(left,r1,r2,r3,r4,r5) private(tmp) reduction(+:s)
225   #else
226   !$omp parallel do private(tmp) reduction(+:s)
227   #endif
233        abssum = s
234     else
235   #ifdef _OPENACC
236 - !$acc parallel loop independent present(left,r1,r2,r3,r4,r5)
237   #else
238   !$omp parallel do
```

```
26      integer :: i
27      !---------------------------------------------------
28    #ifdef _OPENACC
29  + !$acc parallel loop independent gang vector &
30  + !$acc&         present(left,r1)
31      do i=1,sz
32         left(i) = r1(i)
33      enddo
52      complex :: tmp
53      !---------------------------------------------------
54    #ifdef _OPENACC
55  + !$acc parallel loop independent gang vector &
56  + !$acc&         present(left1,left2,r1) private(tmp)
57      do i=1,sz
58         tmp = r1(i)
59         left1(i) = tmp
91      if (present(abssum)) then
92        s = 0.0
93    #ifdef _OPENACC
94  + !$acc parallel loop independent gang vector &
95  + !$acc&         present(left,r1,r2) private(tmp) reduction(+:s)
96    #else
97    !$omp parallel do private(tmp) reduction(+:s)
98    #endif
104        abssum = s
105     else
106   #ifdef _OPENACC
107 + !$acc parallel loop independent gang vector &
108 + !$acc&         present(left,r1,r2)
109   #else
110   !$omp parallel do
111   #endif
135     if (present(abssum)) then
136       s = 0.0
137   #ifdef _OPENACC
138 + !$acc parallel loop independent gang vector &
139 + !$acc&         present(left,r1,r2,r3) private(tmp) reduction(+:s)
140   #else
141   !$omp parallel do private(tmp) reduction(+:s)
142   #endif
148        abssum = s
149     else
150   #ifdef _OPENACC
151 + !$acc parallel loop independent gang vector &
152 + !$acc&         present(left,r1,r2,r3)
153   #else
154   !$omp parallel do
155   #endif
181     if (present(abssum)) then
182       s = 0.0
183   #ifdef _OPENACC
184 + !$acc parallel loop independent gang vector &
185 + !$acc&         present(left,r1,r2,r3,r4) private(tmp) reduction(+:s)
186   #else
187   !$omp parallel do private(tmp) reduction(+:s)
188   #endif
194        abssum = s
195     else
196   #ifdef _OPENACC
197 + !$acc parallel loop independent gang vector &
198 + !$acc&         present(left,r1,r2,r3,r4)
199   #else
200   !$omp parallel do
201   #endif
229     if (present(abssum)) then
230       s = 0.0
231   #ifdef _OPENACC
232 + !$acc parallel loop independent gang vector &
233 + !$acc&         present(left,r1,r2,r3,r4,r5) private(tmp) reduction(+:s)
234   #else
235   !$omp parallel do private(tmp) reduction(+:s)
236   #endif
242        abssum = s
243     else
244   #ifdef _OPENACC
245 + !$acc parallel loop independent gang vector &
246 + !$acc&         present(left,r1,r2,r3,r4,r5)
247   #else
248   !$omp parallel do
```

```
#endif
    if (present(abssum)) then
      s = 0.0
#ifdef _OPENACC
- !$acc parallel loop independent present(left,r1,r2,r3,r4,r5,r6) private(tmp) reduction(+:s)
#else
!$omp parallel do private(tmp) reduction(+:s)
#endif
      abssum = s
    else
#ifdef _OPENACC
- !$acc parallel loop independent present(left,r1,r2,r3,r4,r5,r6)
#else
!$omp parallel do
#endif
    if (present(abssum)) then
      s = 0.0
#ifdef _OPENACC
- !$acc parallel loop independent present(left,r1,rN) copyin(cN) private(tmp,j) reduction(+:s)
#else
!$omp parallel do private(tmp,j) reduction(+:s)
#endif
      abssum = s
    else
#ifdef _OPENACC
- !$acc parallel loop independent present(left,r1,rN) copyin(cN) private(tmp,j)
#else
!$omp parallel do private(tmp,j)
#endif
    sl = 0.0
    sm = 0.0
#ifdef _OPENACC
- !$acc parallel loop independent present(left,r0,m1,rN) copyin(cN,ecN) private(tmp,tmpl,tmpm,j) reduction(+:sl,sm)
#else
!$omp parallel do private(tmp,tmpl,tmpm,j) reduction(+:sl,sm)
#endif
```

```
#endif
    if (present(abssum)) then
      s = 0.0
#ifdef _OPENACC
+ !$acc parallel loop independent gang vector &
+ !$acc&         present(left,r1,r2,r3,r4,r5,r6) private(tmp) reduction(+:s)
#else
!$omp parallel do private(tmp) reduction(+:s)
#endif
      abssum = s
    else
#ifdef _OPENACC
+ !$acc parallel loop independent gang vector &
+ !$acc&         present(left,r1,r2,r3,r4,r5,r6)
#else
!$omp parallel do
#endif
    if (present(abssum)) then
      s = 0.0
#ifdef _OPENACC
+ !$acc parallel loop independent gang vector &
+ !$acc&         present(left,r1,rN) copyin(cN) private(tmp,j) reduction(+:s)
#else
!$omp parallel do private(tmp,j) reduction(+:s)
#endif
      abssum = s
    else
#ifdef _OPENACC
+ !$acc parallel loop independent gang vector &
+ !$acc&         present(left,r1,rN) copyin(cN) private(tmp,j)
#else
!$omp parallel do private(tmp,j)
#endif
    sl = 0.0
    sm = 0.0
#ifdef _OPENACC
+ !$acc parallel loop independent gang vector &
+ !$acc&         present(left,r0,m1,rN) copyin(cN,ecN) private(tmp,tmpl,tmpm,j) &
+ !$acc&         reduction(+:sl,sm)
#else
!$omp parallel do private(tmp,tmpl,tmpm,j) reduction(+:sl,sm)
#endif
```

3 cgyro/src/cgyro_nl_comm.F90

```
  enddo

#ifdef _OPENACC
- !$acc parallel loop independent present(fpack) if ( (nv_loc*n_theta) < (nsplit*n_toroidal_procs) )
#endif
  do iexch=nv_loc*n_theta+1,nsplit*n_toroidal_procs
    fpack(1:n_radial,1:nt_loc,iexch) = (0.0,0.0)
```

```
  enddo

#ifdef _OPENACC
+ !$acc parallel loop independent gang vector &
+ !$acc&         present(fpack) if ( (nv_loc*n_theta) < (nsplit*n_toroidal_procs) )
#endif
  do iexch=nv_loc*n_theta+1,nsplit*n_toroidal_procs
    fpack(1:n_radial,1:nt_loc,iexch) = (0.0,0.0)
```

9 cgyro/src/cgyro_nl_fftw.gpu.F90

```

  integer :: i

- !$acc parallel loop independent present(v1,v2,v3,v4) private(i)
  do i=1,sz
    v1(i) = 0.0
    v2(i) = 0.0

  integer :: i

- !$acc parallel loop independent present(uvm,uxm,vym,uym,vxm) private(i)
  do i=1,sz
    uvm(i) = (uxm(i)*vym(i)-uym(i)*vxm(i))*inv_nxny
  enddo
  ! NOTE: The FFT will generate an unwanted n=0,p=-nr/2 component
  ! that will be filtered in the main time-stepping loop

- !$acc parallel loop independent collapse(4) private(itor,ix,iy) present(f_nl,fxmany)
  do itm=1,n_toroidal_procs
    do itl=1,nt_loc
      do j=1,nsplit
```

```

  integer :: i

+ !$acc parallel loop independent gang vector &
+ !$acc&         present(v1,v2,v3,v4) private(i)
  do i=1,sz
    v1(i) = 0.0
    v2(i) = 0.0

  integer :: i

+ !$acc parallel loop independent gang vector &
+ !$acc&         present(uvm,uxm,vym,uym,vxm) private(i)
  do i=1,sz
    uvm(i) = (uxm(i)*vym(i)-uym(i)*vxm(i))*inv_nxny
  enddo
  ! NOTE: The FFT will generate an unwanted n=0,p=-nr/2 component
  ! that will be filtered in the main time-stepping loop

+ !$acc parallel loop independent collapse(4) gang vector &
+ !$acc&         private(itor,ix,iy) present(f_nl,fxmany)
  do itm=1,n_toroidal_procs
    do itl=1,nt_loc
      do j=1,nsplit
```

5 cgyro/src/cgyro_parallel_lib.F90

```

    j1 = 1+iproc*nj_loc
    j2 = (1+iproc)*nj_loc
- !$acc parallel loop collapse(4) independent private(j_loc) &
- !$acc&         present(fsendr,fin) present(nproc,nk1,nk2,ni_loc) copyin(j1,j2) default(none)
    do k=1,nproc
     do itor=nk1,nk2
       do j=j1,j2
```

```

    j1 = 1+iproc*nj_loc
    j2 = (1+iproc)*nj_loc
+ !$acc parallel loop collapse(4) gang vector independent private(j_loc) &
+ !$acc&         present(fsendr,fin) present(nproc,nk1,nk2,ni_loc) &
+ !$acc&         copyin(j1,j2) default(none)
    do k=1,nproc
     do itor=nk1,nk2
       do j=j1,j2
```

6 cgyro/src/cgyro_rhs.gpu.f90

```
  if (n_field > 1) then
    call timer_lib_in('str')

- !$acc parallel loop  collapse(3) independent private(iv_loc,is) &
  !$acc&         present(is_v,z,temp,jvec_c) &
  !$acc&         present(nt1,nt2,nv1,nv2,nc) &
  !$acc&         default(none) async(1)
  else
    call timer_lib_in('str_mem')

- !$acc parallel loop  collapse(3) independent private(iv_loc) &
  !$acc&         present(nt1,nt2,nv1,nv2,nc)  &
  !$acc&         default(none) async(1)
    do itor=nt1,nt2
```

```
  if (n_field > 1) then
    call timer_lib_in('str')

+ !$acc parallel loop  collapse(3) independent gang vector &
+ !$acc&         private(iv_loc,is) &
  !$acc&         present(is_v,z,temp,jvec_c) &
  !$acc&         present(nt1,nt2,nv1,nv2,nc) &
  !$acc&         default(none) async(1)
  else
    call timer_lib_in('str_mem')

+ !$acc parallel loop  collapse(3) gang vector &
+ !$acc&         independent private(iv_loc) &
  !$acc&         present(nt1,nt2,nv1,nv2,nc)  &
  !$acc&         default(none) async(1)
    do itor=nt1,nt2
```

2 cgyro/src/cgyro_source.F90

```
    icp = (ir-1+1)*n_theta

#ifdef _OPENACC
- !$acc parallel loop private(j) present(h_x,source)
#else
!$omp parallel do private(j)
#endif
```

```
    icp = (ir-1+1)*n_theta

#ifdef _OPENACC
+ !$acc parallel loop gang vector private(j) present(h_x,source)
#else
!$omp parallel do private(j)
#endif
```

4 cgyro/src/cgyro_upwind.F90

```
  call timer_lib_in('str')

#ifdef _OPENACC
- !$acc parallel loop collapse(3) independent &
!$acc&         present(is_v,ix_v,ie_v,xi,vel,upfac2,g_x,upwind_res) &
!$acc&         private(iv_loc,is,ix,ie) present(nt1,nt2,nv1,nv2,nc) default(none)
#else
  call timer_lib_in('str')

#ifdef _OPENACC
- !$acc parallel loop collapse(3) independent &
!$acc&         present(is_v,ix_v,ie_v,xi,vel,upfac2,g_x,upwind32_res) &
!$acc&         private(iv_loc,is,ix,ie) &
!$acc&         present(nt1,nt2,nv1,nv2,nc) default(none)
```

```
  call timer_lib_in('str')

#ifdef _OPENACC
+ !$acc parallel loop collapse(3) independent gang vector &
!$acc&         present(is_v,ix_v,ie_v,xi,vel,upfac2,g_x,upwind_res) &
!$acc&         private(iv_loc,is,ix,ie) present(nt1,nt2,nv1,nv2,nc) default(none)
#else
  call timer_lib_in('str')

#ifdef _OPENACC
+ !$acc parallel loop collapse(3) independent gang vector &
!$acc&         present(is_v,ix_v,ie_v,xi,vel,upfac2,g_x,upwind32_res) &
!$acc&         private(iv_loc,is,ix,ie) &
!$acc&         present(nt1,nt2,nv1,nv2,nc) default(none)
```