2016 年 2 月 9 日

十万次元規模のエルミート行列を用いた連立線形方程式を解くプログラム

依頼内容
原子核の励起状態の計算のため、標題のプログラムが欲しい。

対応
エルミート性をあらわに用いた上記のプログラムは一般に入手可能なライブラリにないため、ユーザー支援チームが共役勾配法を用いて、Fortran90-MPI-OpenMP のプログラムを開発した。

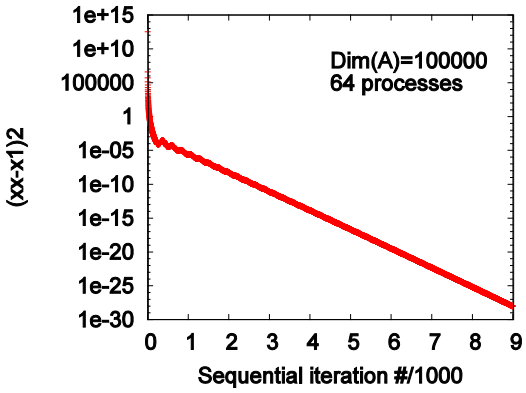

図 1．反復過程で引き続いて求まる二つの途中解ベクトルの差のノルムの二乗を反復番号の関数として示したもの。

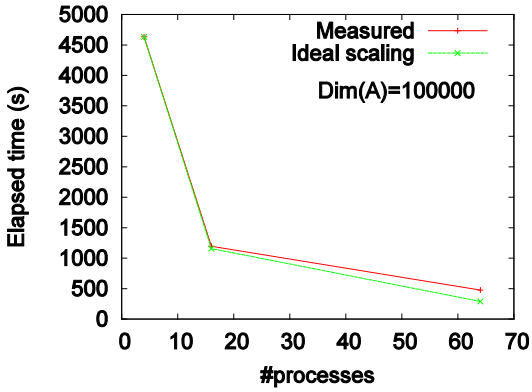

図 2. 実計算時間と用いたプロセス数の関係。理想値は、プロセス数 4 を基点に求めた。

三重帯対角行列をサンプルに用いてテストを行い、図 1 と 2 に示すような性能を得た。1 プロセスは 16 スレッドから成る。最終的に得た解に対し、エルミート行列と解ベクトルの積と定ベクトルの差のノルムの二乗は 10^{-10} であった。このテストには、Fujitsu FX10 を用い、最速計算は 64 プロセスの時の約 8 分であった。次のページ以下にこの計算に用いたプログラムを示す。

```
! cg_omp_mpi2.1, Jan. 23, 2016
! Synclonization strengthened.

program conjugate_gradient
  !$ use omp_lib
  implicit none

  include 'mpif.h'

  integer, parameter :: dp=kind(1.0d0)
  integer, parameter :: mxk=100000

! real(dp), dimension(mxk,mxk) :: a
! real(dp), dimension(mxk) :: x,r,b,p,x1,r1,p1,dv,ap
  real(dp) :: alpha,beta,s,s2,s0,T1,T2
  integer :: i,i1,i2,ic

  integer :: iam
  integer, dimension(100) :: is

  integer, parameter :: nprow=8,npcol=nprow
  integer :: ictxt,nproww,npcolw,myrow,mycol,i1g,i2g,ig,irow,icol
  integer :: mxkcoll,mxkrowl

  integer, parameter :: lda=mxk/nprow+1
  real(dp), dimension(lda,lda) :: a
  real(dp), dimension(lda) :: x,r,b,p,x1,r1,p1,dv,ap,sv,srcv,svoths

  integer :: ierr,nprocs,myrank,mythread,ithread,irowsrc,icolsrc

  logical :: cntl
  real(dp) :: soths,s_rcv,apx
  real(dp), dimension(lda) :: apoths,ap_rcv,xx,pp
  integer :: mxkcoll_rcv

  nproww = nprow
  npcolw = npcol
  call sl_init(ictxt,nproww,npcolw)
  call blacs_gridinfo(ictxt,nproww,npcolw,myrow,mycol)

  myrank = myrow*nprow+mycol
! call mpi_init(ierr)
! if ( ierr /= 0 ) write(6,'(a13,i6)') 'mpi_init,ierr',ierr
! call mpi_comm_size(mpi_comm_world,nprocs,ierr)
! if ( ierr /= 0 ) write(6,'(a18,i6)') 'mpi_comm_size,ierr',ierr
! call mpi_comm_rank(mpi_comm_world,myrank,ierr)
! if ( ierr /= 0 ) write(6,'(a18,i6)') 'mpi_comm_rank,ierr',ierr

  !================================================
  !$OMP PARALLEL
  !================================================

  mythread = omp_get_thread_num()

! mxkcoll : mxk_column_local
! mxkrowl : mxk_row_local

  mxkcoll = mxk/npcolw
  if ( mycol+1 <= mod(mxk,npcolw) ) mxkcoll = mxkcoll +1
  mxkrowl = mxk/npcolw
  if ( myrow+1 <= mod(mxk,nproww) ) mxkrowl = mxkrowl +1

! Sample input
! Vector b is distributed to columns, and every row has the same b.
```

```
  !$OMP DO
  do i=1,mxkcoll
   if ( mycol+1 <= mod(mxk,npcolw) ) then
    ig = mycol*(mxk/npcolw+1) +i
   else
    ig = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i
   endif
   b(i) = ig*0.1_dp
  enddo
  !$OMP END DO
  !$OMP BARRIER

  if ( myrow == 0 ) then
  !$OMP SINGLE
  do i=1,mxkcoll
   if ( mycol+1 <= mod(mxk,npcolw) ) then
    ig = mycol*(mxk/npcolw+1) +i
   else
    ig = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i
   endif
   if ( ig == 100 ) then
    write(6,'(a2,i6,a12,2i6,x,a1,i6,x,a2,d22.14)') 'ig',ig,'
myrow,mycol',myrow,mycol,'i',i,'b ',b(i)
   endif
  enddo
  !$OMP END SINGLE
  endif
  !$OMP BARRIER

  !$OMP DO PRIVATE(i2)
  do i1=1,mxkcoll
     do i2=1,mxkrowl
        a(i2,i1) = 0.0D0
     end do
  end do
  !$OMP END DO
  !$OMP BARRIER
  !$OMP DO PRIVATE(i2,i1g,i2g)
  do i1=1,mxkrowl
     do i2=1,mxkcoll
!         if ( i1 == i2 ) a(i1,i2) = i*0.1_dp
!         if ( i2 == i1 +1 ) a(i1,i2) = i*0.05_dp
! i1g : i1_global
! i2g : i2_global
      if ( myrow+1 <= mod(mxk,nproww) ) then
       i1g = myrow*(mxk/nproww+1) +i1
      else
       i1g = mod(mxk,nproww)*(mxk/nproww+1) +( myrow -mod(mxk,nproww))*(mxk/nproww) +i1
      endif
      if ( mycol+1 <= mod(mxk,npcolw) ) then
       i2g = mycol*(mxk/npcolw+1) +i1
      else
       i2g = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i2
      endif
           if ( i1g == i2g )    a(i1,i2) = i1g*0.1_dp
           if ( i2g == i1g +1 ) a(i1,i2) = i1g*0.05_dp
           if ( i2g == i1g -1 ) a(i1,i2) = (i1g-1)*0.05_dp
     enddo
  enddo
  !$OMP END DO
  !$OMP BARRIER
 !!$OMP DO PRIVATE(i2)
! do i1=2,mxk
!    do i2=1,i1-1
```

```
!       a(i1,i2) = a(i2,i1)
!    enddo
! enddo
 !!$OMP END DO

 !!$OMP BARRIER
  T1 = OMP_GET_WTIME()

  !$OMP BARRIER

  if ( myrow == mycol ) then
  !$OMP DO
  do i=1,mxkcoll
     dv(i) = 1.0_dp/a(i,i)
  enddo
  !$OMP END DO
  endif

  !$OMP BARRIER
 !!$OMP END PARALLEL
  !$OMP MASTER

  call mpi_barrier(mpi_comm_world,ierr)
! By doing the following every row has the same vector dv distributed to columns.
  if ( myrow == mycol ) then
   call dgebs2d( ictxt,'C',' ',mxkcoll,1,dv,lda )
  else
   call dgebr2d( ictxt,'C',' ',mxkcoll,1,dv,lda,mycol,mycol )
  endif
  call mpi_barrier(mpi_comm_world,ierr)

    do i=1,mxkcoll
     if ( mycol+1 <= mod(mxk,npcolw) ) then
      ig = mycol*(mxk/npcolw+1) +i
     else
      ig = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i
     endif
     if ( ig == 100 ) then
      write(6,'(a2,i6,a12,2i6,x,a1,i6,x,a2,d22.14)') 'ig',ig,'
myrow,mycol',myrow,mycol,'i',i,'dv',dv(i)
     endif
    enddo

  !$OMP END MASTER
  !$OMP BARRIER

 !!$OMP PARALLEL
  !$OMP BARRIER
! Initial x (guess)
! Every column has the same vector x distributed to rows
  !$OMP DO PRIVATE(ig)
  do i=1,mxkrowl
   if ( myrow+1 <= mod(mxk,nproww) ) then
    ig = myrow*(mxk/nproww+1) +i
   else
    ig = mod(mxk,nproww)*(mxk/nproww+1) +( myrow -mod(mxk,nproww))*(mxk/nproww) +i
   endif
    x(i) = ig*0.1_dp
  enddo
  !$OMP END DO
  !$OMP BARRIER

  !$OMP SINGLE
  do i=1,mxkrowl
```

```
   if ( myrow+1 <= mod(mxk,nproww) ) then
    ig = myrow*(mxk/nproww+1) +i
   else
    ig = mod(mxk,nproww)*(mxk/nproww+1) +( myrow -mod(mxk,nproww))*(mxk/nproww) +i
   endif
   if ( ig == 100 ) then
    write(6,'(a2,i6,a12,2i6,x,a1,i6,x,a2,d22.14)') 'ig',ig,'
myrow,mycol',myrow,mycol,'i',i,'x ',x(i)
   endif
  enddo
  !$OMP END SINGLE
  !$OMP BARRIER

! Initial r and p
  !$OMP SINGLE
  s0 = 0.0_dp
  ic = 0
  !$OMP END SINGLE
  !$OMP BARRIER

! Vector r is distributed to columns, and every row has the same r.
 !$OMP DO PRIVATE(i1)
  do i=1,mxkcoll
     sv(i) = 0.0_dp
     do i1=1,mxkrowl
!        s = s +a(i,i1)*x(i1)
       sv(i) = sv(i) +a(i1,i)*x(i1)
     enddo
  enddo
 !$OMP END DO
 !$OMP BARRIER
 !$OMP MASTER
     svoths(:) = 0.0_dp
     do irow=1,nproww
      if ( myrow+1 == irow ) then
       call dgebs2d( ictxt,'C',' ',lda,1,sv  ,lda )
      else
       irowsrc = irow -1
       call dgebr2d( ictxt,'C',' ',lda,1,srcv,lda,irowsrc,mycol )
       do i=1,mxkcoll
        svoths(i) = svoths(i) +srcv(i)
       enddo
      endif
      call mpi_barrier(mpi_comm_world,ierr)
     enddo ! irow
     do i=1,mxkcoll
      sv(i) = sv(i) +svoths(i)
     enddo
 !$OMP END MASTER
 !$OMP BARRIER
 !$OMP DO
  do i=1,mxkcoll
     r(i) = b(i) -sv(i)
  enddo
  !$OMP END DO
  !$OMP BARRIER
       !$OMP SINGLE
       do i=1,mxkcoll
        if ( mycol+1 <= mod(mxk,npcolw) ) then
         ig = mycol*(mxk/npcolw+1) +i
        else
         ig = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i
        endif
        if ( ig == 100 ) then
```

```
         write(6,'(a2,i6,a12,2i6,x,a1,i6,x,a2,d22.14)') 'ig',ig,'
myrow,mycol',myrow,mycol,'i',i,'sv',sv(i)
         write(6,'(a2,i6,a12,2i6,x,a1,i6,x,a2,d22.14)') 'ig',ig,'
myrow,mycol',myrow,mycol,'i',i,'r ',r(i)
        endif
       enddo
       !$OMP END SINGLE
 !$OMP BARRIER

 !$OMP BARRIER
mythread = omp_get_thread_num()
write(7,'(a12,x,a11,2i6,x,a8,i6)') 'I came
here!','myrow,mycol',myrow,mycol,'mythread',mythread

!  p = r
  !$OMP DO
! do i=1,mxk
  do i=1,mxkcoll
     p(i) = dv(i) * r(i)
  end do
  !$OMP END DO
  !$OMP BARRIER

!10 &
10 continue

  !$OMP SINGLE
  ic = ic +1
  !$OMP END SINGLE
  !$OMP BARRIER

!  s = 0.0_dp
  !$OMP SINGLE
  s = 0.0_dp
  !$OMP END SINGLE
  !$OMP BARRIER
 !!$OMP DO PRIVATE(i) REDUCTION(+:s)
  !$OMP DO            REDUCTION(+:s)
! do i=1,mxk
  do i=1,mxkcoll
     s = s +r(i)*dv(i)*r(i)
  enddo
  !$OMP END DO
  !$OMP BARRIER
 !$OMP MASTER
  call mpi_barrier(mpi_comm_world,ierr)
 !$OMP END MASTER

 !$OMP MASTER
  soths = 0.0_dp
  do icol=1,npcolw
   if ( mycol+1 == icol ) then
    call dgebs2d( ictxt,'R',' ',1,1,s  ,1 )
   else
    icolsrc = icol -1
    call dgebr2d( ictxt,'R',' ',1,1,s_rcv,1,myrow,icolsrc )
    soths = soths +s_rcv
!   write(6,'(a9,2i9,a6,d22.14,a6,d22.14)') 'myrow,col',myrow,mycol,' s_rcv',s_rcv,'
soths',soths
   endif
  enddo ! irow
! write(6,'(a9,2i9,a6,a6,d22.14          )') 'myrow,col',myrow,mycol,  ' soths',    '1
s',s
  call mpi_barrier(mpi_comm_world,ierr)
```

```
  s = s +soths
! write(6,'(a9,2i9,a6,a6,d22.14          )') 'myrow,col',myrow,mycol,  ' soths',    '2
s',s
 !$OMP END MASTER
 !$OMP BARRIER

  !$OMP SINGLE
  write(6,'(a9,2i9,a9,d22.14)') 'myrow,col',myrow,mycol,' s,r*dv*r',s
  !$OMP END SINGLE

  !$OMP SINGLE
  s2 = 0.0_dp
  !$OMP END SINGLE
  !$OMP BARRIER

  !$OMP BARRIER
  !$OMP MASTER
  if ( myrow == mycol ) then
    call dgebs2d( ictxt,'R',' ',lda,1,p,lda )
    pp = p
  else
    call dgebr2d( ictxt,'R',' ',lda,1,pp,lda,myrow,myrow )
  endif
  call mpi_barrier(mpi_comm_world,ierr)
  !$OMP END MASTER
  !$OMP BARRIER


  !$OMP DO PRIVATE(i2) REDUCTION(+:apx)
! do i1=1,mxk
  do i1=1,mxkcoll
!    ap(i1) = 0.0_dp
     apx = 0.0_dp
     do i2=1,mxkrowl
!        ap(i1) = ap(i1) +a(i1,i2)*p(i2)
!       ap(i1) = ap(i1) +a(i2,i1)*p(i2)
!       apx    = apx    +a(i2,i1)*p(i2)
        apx    = apx    +a(i2,i1)*pp(i2)
     enddo
     ap(i1) = apx
  enddo
!$OMP END DO
!$OMP BARRIER
!$OMP MASTER
      call mpi_barrier(mpi_comm_world,ierr)
     apoths(:) = 0.0_dp
     do irow=1,nproww
      if ( myrow+1 == irow ) then
       call dgebs2d( ictxt,'C',' ',lda,1,ap  ,lda )
      else
       irowsrc = irow -1
       call dgebr2d( ictxt,'C',' ',lda,1,ap_rcv,lda,irowsrc,mycol )
       do i=1,mxkcoll
        apoths(i) = apoths(i) +ap_rcv(i)
       enddo
      endif
     enddo ! irow
     call mpi_barrier(mpi_comm_world,ierr)
     do i=1,mxkcoll
      ap(i) = ap(i) +apoths(i)
     enddo
 !$OMP END MASTER
 !$OMP BARRIER
```

```
  !$OMP SINGLE
   do i=1,mxkcoll
    if ( mycol+1 <= mod(mxk,npcolw) ) then
     ig = mycol*(mxk/npcolw+1) +i
    else
     ig = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i
    endif
    if ( ig == 10 .or. ig == 60 ) then
     write(6,'(a2,i6,a12,2i6,x,a1,i6,x,a2,d22.14)') 'ig',ig,'
myrow,mycol',myrow,mycol,'i',i,'p ',p(i)
     write(6,'(a2,i6,a12,2i6,x,a1,i6,x,a2,d22.14)') 'ig',ig,'
myrow,mycol',myrow,mycol,'i',i,'ap',ap(i)
    endif
   enddo
  !$OMP END SINGLE
  !$OMP BARRIER

  !$OMP DO REDUCTION(+:s2)
! do i1=1,mxk
  do i1=1,mxkcoll
     s2 = s2 +p(i1)*ap(i1)
!    s2 = s2 +p(i1)
!    s2 = s2 +      ap(i1)
  enddo
  !$OMP END DO
  !$OMP BARRIER
  !$OMP MASTER
  call mpi_barrier(mpi_comm_world,ierr)
! write(myrank+11,'(a9,2i6,a3,d22.14,a8,i6)') 'myrow,col',myrow,mycol,' s2',s2,'
mxkcoll',mxkcoll
  soths = 0.0_dp
  do icol=1,npcolw
   if ( mycol+1 == icol ) then
    call dgebs2d( ictxt,'R',' ',1,1,s2,1 )
   else
    icolsrc = icol -1
    call dgebr2d( ictxt,'R',' ',1,1,s_rcv,1,myrow,icolsrc )
    soths = soths +s_rcv
   endif
  enddo ! irow
  call mpi_barrier(mpi_comm_world,ierr)
  s2 = s2 +soths
 !$OMP END MASTER
 !$OMP BARRIER

  !$OMP SINGLE
  write(6,'(a9,2i9,a9,d22.14)') 'myrow,col',myrow,mycol,' s2,p*ap ',s2
  !$OMP END SINGLE
  !$OMP BARRIER

  alpha = s/s2

  !$OMP SINGLE
  write(6,'(a9,2i6,a9,d22.14)') 'myrow,col',myrow,mycol,' alpha   ',alpha
  !$OMP END SINGLE

! xx : vector distributed in a column, and every column has the same vector.
  !$OMP BARRIER
  !$OMP MASTER
  if ( myrow == mycol ) then
    call dgebs2d( ictxt,'C',' ',lda,1,x,lda )
    xx = x
  else
    call dgebr2d( ictxt,'C',' ',lda,1,xx,lda,mycol,mycol )
```

```
  endif
  call mpi_barrier(mpi_comm_world,ierr)
  !$OMP END MASTER
  !$OMP BARRIER

! x : distributed in a row
! x1, r1, p, r, ap, and xx : distributed in a column
! if ( myrow == mycol ) then
  !$OMP DO
  do i=1,mxkcoll
!    x1(i) = x(i) +alpha*p(i)
     x1(i) = xx(i) +alpha*p(i)
  enddo
  !$OMP END DO
! endif
  !$OMP BARRIER

  !$OMP SINGLE
   do i=1,mxkcoll
    if ( mycol+1 <= mod(mxk,npcolw) ) then
     ig = mycol*(mxk/npcolw+1) +i
    else
     ig = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i
    endif
    if ( ig == 10 .or. ig == 60 ) then
     write(6,'(a2,i6,a12,2i6,x,a1,i6,a3,d22.14,a6,d22.14,a3,d22.14,x,a2,d22.14)') &
      'ig',ig,' myrow,mycol',myrow,mycol,'i',i,' xx',xx(i),' alpha',alpha,' p
',p(i),'x1',x1(i)
    endif
   enddo
  !$OMP END SINGLE
  !$OMP BARRIER

  !$OMP DO
  do i=1,mxkcoll
     r1(i) = r(i) -alpha*ap(i)
  enddo
  !$OMP END DO
  !$OMP BARRIER

  !$OMP SINGLE
   do i=1,mxkcoll
    if ( mycol+1 <= mod(mxk,npcolw) ) then
     ig = mycol*(mxk/npcolw+1) +i
    else
     ig = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i
    endif
    if ( ig == 10 .or. ig == 60 ) then
     write(6,'(a2,i6,a12,2i6,x,a1,i6,a3,d22.14,a6,d22.14,x,a2,d22.14,a3,d22.14)') &
      'ig',ig,' myrow,mycol',myrow,mycol,'i',i,' r ',r(i),' alpha',alpha,'ap',ap(i),'
r1',r1(i)
    endif
   enddo
  !$OMP END SINGLE
  !$OMP BARRIER

  !$OMP SINGLE
  s = 0.0_dp
  s2 = 0.0_dp
  !$OMP END SINGLE
  !$OMP BARRIER
  !$OMP DO REDUCTION(+:s,s2)
  do i=1,mxkcoll
     s = s +r1(i)*dv(i)*r1(i)
```

```
      s2 = s2 +r(i)*dv(i)*r(i)
    enddo
    !$OMP END DO
    !$OMP BARRIER

    !$OMP MASTER
    call mpi_barrier(mpi_comm_world,ierr)
    soths = 0.0_dp
    do icol=1,npcolw
     if ( mycol+1 == icol ) then
      call dgebs2d( ictxt,'R',' ',1,1,s  ,1 )
     else
      icolsrc = icol -1
      call dgebr2d( ictxt,'R',' ',1,1,s_rcv,1,myrow,icolsrc )
      soths = soths +s_rcv
     endif
    enddo ! irow
    call mpi_barrier(mpi_comm_world,ierr)
    s = s +soths
   !$OMP END MASTER
   !$OMP BARRIER

    !$OMP MASTER
    soths = 0.0_dp
    do icol=1,npcolw
     if ( mycol+1 == icol ) then
      call dgebs2d( ictxt,'R',' ',1,1,s2 ,1 )
     else
      icolsrc = icol -1
      call dgebr2d( ictxt,'R',' ',1,1,s_rcv,1,myrow,icolsrc )
      soths = soths +s_rcv
     endif
    enddo ! irow
    call mpi_barrier(mpi_comm_world,ierr)
    s2 = s2 +soths
   !$OMP END MASTER
   !$OMP BARRIER

    !$OMP SINGLE
    beta = s/s2
    !$OMP END SINGLE
    !$OMP BARRIER

    !$OMP SINGLE
    write(6,'(a9,2i6,a11,3e22.14)') 'myrow,col',myrow,mycol,' beta, s,s2',beta,s,s2
    !$OMP END SINGLE
    !$OMP BARRIER

    !$OMP DO
    do i=1,mxkcoll
       p1(i) = dv(i)*r1(i) +beta*p(i)
    enddo
    !$OMP END DO
    !$OMP BARRIER

    !$OMP SINGLE
    s = 0.0_dp
    !$OMP END SINGLE
    !$OMP BARRIER
    !$OMP DO REDUCTION(+:s)
    do i=1,mxkcoll
!    s = s +( x(i) -x1(i) )**2
     s = s +( xx(i) -x1(i) )**2
    enddo
```

```
  !$OMP END DO
  !$OMP BARRIER

  !$OMP MASTER
  soths = 0.0_dp
  do icol=1,npcolw
   if ( mycol+1 == icol ) then
    call dgebs2d( ictxt,'R',' ',1,1,s  ,1 )
   else
    icolsrc = icol -1
    call dgebr2d( ictxt,'R',' ',1,1,s_rcv,1,myrow,icolsrc )
    soths = soths +s_rcv
   endif
  enddo ! irow
  call mpi_barrier(mpi_comm_world,ierr)
  s = s +soths
 !$OMP END MASTER
 !$OMP BARRIER

  !$OMP SINGLE
  write(6,'(a9,2i6,a11,3e22.14)') 'myrow,col',myrow,mycol,' s,(xx-x1)2',s
  !$OMP END SINGLE
  !$OMP BARRIER

! if ( ic > 1 .and. s-s0 > 0.0e+0_dp ) goto 20
  if ( s < 1.0e-28_dp ) goto 20
  !$OMP BARRIER

!   x = x1
!   r = r1
!   p = p1
  if ( myrow == mycol ) then
  !$OMP DO
  do i=1,mxkcoll
     x(i) = x1(i)
!    x(i) = 0.9d0*x(i) +0.1d0*x1(i)
  end do
  !$OMP END DO
  endif
  !$OMP BARRIER
  !$OMP MASTER
  if ( myrow == mycol ) then
    call dgebs2d( ictxt,'R',' ',lda,1,x,lda )
  else
    call dgebr2d( ictxt,'R',' ',lda,1,x,lda,myrow,myrow )
  endif
  xx = x1
  call mpi_barrier(mpi_comm_world,ierr)
  !$OMP END MASTER
  !$OMP BARRIER

  !$OMP DO
  do i=1,mxkcoll
     r(i) = r1(i)
     p(i) = p1(i)
!    r(i) = 0.9d0*r(i) +0.1d0*r1(i)
!    p(i) = 0.9d0*p(i) +0.1d0*p1(i)
  end do
  !$OMP END DO
  !$OMP BARRIER

  !$OMP SINGLE
   do i=1,mxkcoll
    if ( mycol+1 <= mod(mxk,npcolw) ) then
```

```
      ig = mycol*(mxk/npcolw+1) +i
     else
      ig = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i
     endif
     if ( ig == 10 .or. ig == 60 ) then
      write(6,'(a2,i6,a12,2i6,x,a1,i6,a6,d22.14,a3,d22.14)') &
       'ig',ig,' myrow,mycol',myrow,mycol,'i',i,' r    ',r(i),' p ',p(i)
     endif
    enddo
    do i=1,mxkrowl
     if ( myrow+1 <= mod(mxk,nproww) ) then
      ig = myrow*(mxk/nproww+1) +i
     else
      ig = mod(mxk,nproww)*(mxk/nproww+1) +( myrow -mod(mxk,nproww))*(mxk/nproww) +i
     endif
     if ( ig == 10 .or. ig == 60 ) then
      write(6,'(a2,i6,a12,2i6,x,a1,i6,a3,d22.14)') &
       'ig',ig,' myrow,mycol',myrow,mycol,'i',i,' x ',x(i)
     endif
    enddo
    do i=1,mxkrowl
     if ( myrow+1 <= mod(mxk,nproww) ) then
      ig = myrow*(mxk/nproww+1) +i
     else
      ig = mod(mxk,nproww)*(mxk/nproww+1) +( myrow -mod(mxk,nproww))*(mxk/nproww) +i
     endif
     if ( ig == 10 .or. ig == 60 ) then
      write(6,'(a2,i6,a12,2i6,x,a1,i6,a3,d22.14)') &
       'ig',ig,' myrow,mycol',myrow,mycol,'i',i,' x ',x(i)
     endif
    enddo
   !$OMP END SINGLE
   !$OMP BARRIER

! x = x*0.7_dp +x1*0.3_dp
! r = r*0.3_dp +r1*0.7_dp
! p = p*0.3_dp +p1*0.7_dp

   !$OMP SINGLE
   s0 = s
   !$OMP END SINGLE
   !$OMP BARRIER
   !$OMP MASTER
   call mpi_barrier(mpi_comm_world,ierr)
   !$OMP END MASTER
   !$OMP BARRIER

   goto 10

!20 &
20 continue
!  write(6,'(a7,e14.6)') 'dxdx   ',s

   !$OMP BARRIER
   T2 = OMP_GET_WTIME()

   !$OMP DO PRIVATE(i2)
   do i1=1,mxkcoll
      x1(i1) = 0.0_dp
      do i2=1,mxkrowl
!        x1(i1) = x1(i1) +a(i1,i2)*xx(i2)
         x1(i1) = x1(i1) +a(i2,i1)*x(i2)
      enddo
   enddo
```

```
  !$OMP END DO
 !$OMP BARRIER
 !$OMP MASTER
     apoths(:) = 0.0_dp
     do irow=1,nproww
      if ( myrow+1 == irow ) then
       call dgebs2d( ictxt,'C',' ',lda,1,x1  ,lda )
      else
       irowsrc = irow -1
       call dgebr2d( ictxt,'C',' ',lda,1,ap_rcv,lda,irowsrc,mycol )
       do i=1,mxkcoll
        apoths(i) = apoths(i) +ap_rcv(i)
       enddo
      endif
!     call mpi_barrier(mpi_comm_world,ierr)
     enddo ! irow
     do i=1,mxkcoll
      x1(i) = x1(i) +apoths(i)
     enddo
  call mpi_barrier(mpi_comm_world,ierr)
 !$OMP END MASTER
 !$OMP BARRIER

  !$OMP SINGLE
   do i=1,mxkcoll
    if ( mycol+1 <= mod(mxk,npcolw) ) then
     ig = mycol*(mxk/npcolw+1) +i
    else
     ig = mod(mxk,npcolw)*(mxk/npcolw+1) +( mycol -mod(mxk,npcolw))*(mxk/npcolw) +i
    endif
    if ( ig == 10 .or. ig == 60 ) then
     write(6,'(a2,i6,a12,2i6,x,a1,i6,a3,d22.14)') &
      'ig',ig,' myrow,mycol',myrow,mycol,'i',i,' x1 ',x1(i)
    endif
   enddo
  !$OMP END SINGLE
  !$OMP BARRIER

  !$OMP SINGLE
  s = 0.0_dp
  s2 = 0.0_dp
  !$OMP END SINGLE
  !$OMP BARRIER
  !$OMP DO REDUCTION(+:s,s2)
  do i=1,mxkcoll
     s = s +b(i)*b(i)
     s2 = s2 +( x1(i) -b(i) )**2
  enddo
  !$OMP END DO
  !$OMP BARRIER

  !$OMP MASTER
  soths = 0.0_dp
  do icol=1,npcolw
   if ( mycol+1 == icol ) then
    call dgebs2d( ictxt,'R',' ',1,1,s  ,1 )
   else
    icolsrc = icol -1
    call dgebr2d( ictxt,'R',' ',1,1,s_rcv,1,myrow,icolsrc )
    soths = soths +s_rcv
   endif
  enddo ! irow
  call mpi_barrier(mpi_comm_world,ierr)
  s = s +soths
```

```
  soths = 0.0_dp
  do icol=1,npcolw
   if ( mycol+1 == icol ) then
    call dgebs2d( ictxt,'R',' ',1,1,s2 ,1 )
   else
    icolsrc = icol -1
    call dgebr2d( ictxt,'R',' ',1,1,s_rcv,1,myrow,icolsrc )
    soths = soths +s_rcv
   endif
  enddo ! irow
  call mpi_barrier(mpi_comm_world,ierr)
  s2 = s2 +soths
 !$OMP END MASTER
 !$OMP BARRIER

  !$OMP SINGLE
  write(6,'(a7,e22.14)') 'bb     ',s
  write(6,'(a7,e22.14)') '(Ax-b)2',s2

  write(6,'(x)')
!   do i=1,mxk
!    write(6,'(2d22.14)') x1(i),b(i)
!   enddo
  write(*,'("#Iterations: ",i6            )') ic
  write(*,'("Time       : ",f12.7," [sec]")') T2-T1
  write(*,'("Time/#Iter.: ",f12.7," [sec]")') (T2-T1)/dble(ic)
  !$OMP END SINGLE
  !$OMP BARRIER

 !$OMP MASTER
     apoths(:) = 0.0_dp
     if ( mycol == 0 ) then

     open(10,file='x',status='unknown')
     if ( myrow == 0 ) then
      do i=1,mxkcoll
       write(10,'(d22.14)') x(i)
      enddo
     endif
     do irow=2,nproww
      if ( myrow+1 == irow ) then
       call dgebs2d( ictxt,'C',' ',lda,1,x      ,lda )
       call igebs2d( ictxt,'C',' ',1  ,1,mxkcoll,1 )
      else
       irowsrc = irow -1
       call dgebr2d( ictxt,'C',' ',lda,1,ap_rcv     ,lda,irowsrc,mycol )
       call igebr2d( ictxt,'C',' ',1  ,1,mxkcoll_rcv,1  ,irowsrc,mycol )
       if ( myrow == 0 ) then
        do i=1,mxkcoll_rcv
         write(10,'(d22.14)') ap_rcv(i)
        enddo
       endif ! myrow=0
      endif
     enddo ! irow
     close(10)

     endif ! only mycol=0
     call mpi_barrier(mpi_comm_world,ierr)
 !$OMP END MASTER
 !$OMP BARRIER

 !$OMP END PARALLEL
  goto 111
```

```
 !$OMP PARALLEL

  !================================================
  !$OMP END PARALLEL
  !================================================

111 &
  continue
! call mpi_finalize(ierr)
  call blacs_gridexit(ictxt)
  call blacs_exit(0)

end program conjugate_gradient
```