```
!=====================================================================!
!                                                                     !
!   Software Name : HEC-MW Library for PC-cluster                     !
!         Version : 2.5                                               !
!                                                                     !
!     Last Update : 2006/06/01                                        !
!        Category : Linear Solver                                     !
!                                                                     !
!            Written by Kengo Nakajima (Univ. of Tokyo)               !
!                                                                     !
!     Contact address :  IIS,The University of Tokyo RSS21 project   !
!                                                                     !
!     "Structural Analysis System for General-purpose Coupling       !
!      Simulations Using High End Computing Middleware (HEC-MW)"      !
!                                                                     !
!=====================================================================!

!C
!C***
!C*** module hecmw_solver_SR_33
!C***
!C
      module hecmw_solver_SR_33
      contains
!C
!C*** SOLVER_SEND_RECV
!C
      subroutine  HECMW_SOLVE_SEND_RECV_33                             &
     &                 ( N, NEIBPETOT, NEIBPE, STACK_IMPORT, NOD_IMPORT, &
     &                                         STACK_EXPORT, NOD_EXPORT, &
     &                   WS, WR, X, SOLVER_COMM,my_rank)

      use hecmw_util
      implicit REAL*8 (A-H,O-Z)
!      include  'mpif.h'
!      include  'hecmw_config_f.h'
```

```
37
38          integer(kind=kint )                , intent(in)   ::  N
39          integer(kind=kint )                , intent(in)   ::  NEIBPETOT
40          integer(kind=kint ), pointer :: NEIBPE      (:)
41          integer(kind=kint ), pointer :: STACK_IMPORT(:)
42          integer(kind=kint ), pointer :: NOD_IMPORT  (:)
43          integer(kind=kint ), pointer :: STACK_EXPORT(:)
44          integer(kind=kint ), pointer :: NOD_EXPORT  (:)
45          real   (kind=kreal), dimension(3*N), intent(inout):: WS
46          real   (kind=kreal), dimension(3*N), intent(inout):: WR
47          real   (kind=kreal), dimension(3*N), intent(inout):: X
48          integer(kind=kint )                , intent(in)   ::SOLVER_COMM
49          integer(kind=kint )                , intent(in)   :: my_rank
50
51          integer(kind=kint ), dimension(:,:), allocatable :: sta1
52          integer(kind=kint ), dimension(:,:), allocatable :: sta2
53          integer(kind=kint ), dimension(:  ), allocatable :: req1
54          integer(kind=kint ), dimension(:  ), allocatable :: req2
55
56          integer(kind=kint ), save :: NFLAG
57          data NFLAG/0/
58
59          ! local valiables
60          integer(kind=kint ) :: neib,istart,inum,k,ii,ierr
61    !C
62    !C-- INIT.
63          allocate (sta1(MPI_STATUS_SIZE,NEIBPETOT))
64          allocate (sta2(MPI_STATUS_SIZE,NEIBPETOT))
65          allocate (req1(NEIBPETOT))
66          allocate (req2(NEIBPETOT))
67
68    !C
69    !C-- SEND
70          do neib= 1, NEIBPETOT
71            istart= STACK_EXPORT(neib-1)
72            inum  = STACK_EXPORT(neib  ) - istart
```

```
          do k= istart+1, istart+inum
                ii   = 3*NOD_EXPORT(k)
             WS(3*k-2)= X(ii-2)
             WS(3*k-1)= X(ii-1)
             WS(3*k  )= X(ii  )
          enddo

          call MPI_ISEND (WS(3*istart+1), 3*inum,MPI_DOUBLE_PRECISION,    &
        &                 NEIBPE(neib), 0, SOLVER_COMM, req1(neib), ierr)
        enddo

!C
!C-- RECEIVE
        do neib= 1, NEIBPETOT
          istart= STACK_IMPORT(neib-1)
          inum  = STACK_IMPORT(neib  ) - istart
          call MPI_IRECV (WR(3*istart+1), 3*inum, MPI_DOUBLE_PRECISION,   &
        &                 NEIBPE(neib), 0, SOLVER_COMM, req2(neib), ierr)
        enddo

        call MPI_WAITALL (NEIBPETOT, req2, sta2, ierr)

        do neib= 1, NEIBPETOT
          istart= STACK_IMPORT(neib-1)
          inum  = STACK_IMPORT(neib  ) - istart
        do k= istart+1, istart+inum
            ii   = 3*NOD_IMPORT(k)
          X(ii-2)= WR(3*k-2)
          X(ii-1)= WR(3*k-1)
          X(ii  )= WR(3*k  )
        enddo
        enddo

        call MPI_WAITALL (NEIBPETOT, req1, sta1, ierr)
        deallocate (sta1, sta2, req1, req2)

```

```
109         end subroutine hecmw_solve_send_recv_33
110         end module     hecmw_solver_SR_33
```