```
!
!  Subroutine to set matrix elements for the Laplacian in the reaction-
!  diffusion problem with mixed boundary conditions, discretised using
!  finite differences (five-point stencil), in compressed row storage
!  format (parallel version).
!
!     Note: this uses the standard ordering of the unknowns as presented
!     on the coursework sheet: arranging the indices (i,j) in the order
!
!   (0,0),(1,0),(2,0),...,(m,0),(0,1),...,(m,1),...,(0,m),...,(mdelta,m).
!
!------------------------------------------------------------------------

subroutine Laplace(A,m,ibeg,iend)

  use header
  implicit none

   type(Matrix), intent(inout) :: A
   integer,      intent(in)  :: m,ibeg,iend

   integer :: row,i,j,inz,c
   real(kind=8) :: h

   c = m*m
   h = 1.0d0/m

   A%bw = m+1

   inz = 0
   do row=ibeg,iend

! Calculate the indices (i,j) in the cartesian numbering of
! the unknowns.

      j = (row-1)/(m+1)
      i = row - 1 - j*(m+1)

! Set the diagonal entry and the off-diagonal entries in each
! row according to their position in the mesh.

      if (j == 0) then

         if (i == 0 .or. i == m ) then

            inz = inz + 1
            A%aa(inz) = 2.0d0*c
            A%ii(row) = inz
            A%jj(inz) = row

         else

            inz = inz + 1
            A%aa(inz) = 3.0d0*c
            A%ii(row) = inz
            A%jj(inz) = row

         end if

         if (i > 0) then

            inz = inz + 1
            A%aa(inz) = -c
            A%jj(inz) = row - 1

         end if
      if (i < m) then

        inz = inz + 1
        A%aa(inz) = -c
        A%jj(inz) = row + 1

      end if

      inz = inz + 1
      A%aa(inz) = -c
      A%jj(inz) = row + A%bw

    else if (j < m) then

      if (i == 0 .or. i == m) then

        inz = inz + 1
        A%aa(inz) = 3.0d0*c
        A%ii(row) = inz
        A%jj(inz) = row

      else

        inz = inz + 1
        A%aa(inz) = 4.0d0*c
        A%ii(row) = inz
        A%jj(inz) = row

      end if

      inz = inz + 1
      A%aa(inz) = -c
      A%jj(inz) = row - A%bw

      if (i > 0) then

        inz = inz + 1
        A%aa(inz) = -c
        A%jj(inz) = row - 1

      end if

      if (i < m) then

        inz = inz + 1
        A%aa(inz) = -c
        A%jj(inz) = row + 1

      end if

      if (j < m-1 .or. i < m/2 + 1) then

        inz = inz + 1
        A%aa(inz) = -c
        A%jj(inz) = row + A%bw

      end if

    else

      if (i == 0) then

        inz = inz + 1
        A%aa(inz) = 2.0d0*c
        A%ii(row) = inz
        A%jj(inz) = row
```

```
         else

            inz = inz + 1
            A%aa(inz) = 3.0d0*c
            A%ii(row) = inz
            A%jj(inz) = row

         end if

         inz = inz + 1
         A%aa(inz) = -c
         A%jj(inz) = row - A%bw

         if (i > 0) then

            inz = inz + 1
            A%aa(inz) = -c
            A%jj(inz) = row - 1

         end if

         if (i < m/2) then

            inz = inz + 1
            A%aa(inz) = -c
            A%jj(inz) = row + 1

         end if

      end if

   end do

! Set A%nnz (the number of nonzero entries) and A%ii(A%n+1) which is
! needed to address the nonzero entries in the last row (in order to
! know where the arrays A%aa and A%jj end)

   A%nnz  = inz
   A%ii(A%iend+1) = A%nnz + 1

end subroutine Laplace
```