```
subroutine check_areas(nface, scale)

  ! Check all quadrilateral faces for skewed/irregular or "bad" faces.
  !
  ! 1.0 - 04/24/14 - flc - Stripped from PREP V7.2b.  Moved mesh arrays to
  !                        modules.  Replaced hard-coded bad face coordinate
  !                        arrays with scratch file read & writes, included
  !                        the scale factor in the tecplot writes
  !
  use flag,      only: debug
  use mesh_vars, only: ie_face, ij_edge, ne_face, xnd, ynd, znd
  implicit none

  !--- INPUT
  integer, intent(in) :: nface          ! number of faces
  real(8), intent(in) :: scale          ! grid scaling factor
  !--- LOCAL
  integer :: i, j, k, l                 ! dummy indices
  integer, parameter :: io_bad=100      ! bad face scratch file ID
  integer, parameter :: io_irr=200      ! irregular face scratch file ID
  integer :: edge1, edge2, edge3, edge4 ! face edges
  integer :: nbadface                   ! number of bad faces
  integer :: nirregularface             ! number of irregular faces
  integer :: nvert                      ! number of vertices for tecplot file
  logical :: ok_face, ok_irregular      ! flags indicating good faces
  integer :: node(8)                    ! node index
  real(8) :: x(4), y(4), z(4)           ! face grid coordinates
  !-----------------------------------------------------------------------
  if ( debug ) write(*,*) "calling check_areas"

  open(io_bad, status="scratch", form="unformatted")
  open(io_irr, status="scratch", form="unformatted")
  nbadface = 0; nirregularface = 0
  do i = 1,nface
     if (ne_face(i) .eq. 4) then
        edge1 = ie_face(i, 1)
        edge2 = ie_face(i, 2)
        edge3 = ie_face(i, 3)
        edge4 = ie_face(i, 4)

        node(1) = ij_edge(edge1, 1)
        node(2) = ij_edge(edge1, 2)
        node(3) = ij_edge(edge2, 1)
        node(4) = ij_edge(edge2, 2)
        node(5) = ij_edge(edge3, 1)
        node(6) = ij_edge(edge3, 2)
        node(7) = ij_edge(edge4, 1)
        node(8) = ij_edge(edge4, 2)

        l=1
        do j = 1,7
           do k = j+1,8
              if (node(j) .eq. node(k)) then
                 x(l) = xnd(node(k))
                 y(l) = ynd(node(k))
                 z(l) = znd(node(k))
                 l=l+1
              end if
           end do
        end do

        if (l .ne. 5) then
           write (*,*) "A face does not have 4 nodes"
           stop
        end if

        call check_face(x, y, z, 1, ok_face, ok_irregular)
        !ok_irregular = .true. ! NRM
        !ok_face = .true.
```



```
        if ( .not.ok_irregular ) then
           write (*,*) "Skewed or irregular face: ",i
           nirregularface = nirregularface + 1
           write(io_irr) (x(j), j = 1,4), (y(j), j = 1,4), (z(j), j = 1,4)
        end if

        if ( .not.ok_face ) then
           write (*,*) "Bad face: ",i
           nbadface = nbadface + 1
           write(io_bad) (x(j), j = 1,4), (y(j), j = 1,4), (z(j), j = 1,4)
        end if
     end if
  end do

  ! Write irregularfaces.plt
  if (nirregularface .ne. 0) then
     nvert = 4*nirregularface
     rewind(io_irr)
     write (*,*) "***irregularfaces.plt created***"
     open (50,file = "irregularfaces.plt")
     write (50,*) "variables = x, y, z"
     write (50,*) "ZONE F = FEPOINT, ET = QUADRILATERAL, N = ", nvert, &
          ", E = ",nirregularface
     do i = 1,nirregularface
        read(io_irr) (x(j), j = 1,4), (y(j), j = 1,4), (z(j), j = 1,4)
        do j = 1,4
           write (50,*) scale*x(j), scale*y(j), scale*z(j)
        end do
     end do
     do i = 1,nirregularface
        write (50,*) 4*(i-1)+1,4*(i-1)+2,4*(i-1)+3,4*(i-1)+4
     end do
     close (50)
  end if
  close(io_irr)

  ! Write badfaces.plt
  if (nbadface .ne. 0) then
     rewind(io_bad)
     nvert = 4*nbadface
     write (*,*) "***badfaces.plt created***"
     open (60,file = "badfaces.plt")
     write (60,*) "variables = x, y, z"
     write (60,*) "ZONE F = FEPOINT, ET = QUADRILATERAL, N = ",nvert, &
          ", E = ",nbadface
     do i = 1,nbadface
        read(io_bad) (x(j), j = 1,4), (y(j), j = 1,4), (z(j), j = 1,4)
        do j = 1,4
           write (60,*) scale*x(j), scale*y(j), scale*z(j)
        end do
     end do
     do i = 1,nbadface
        write (60,*) 4*(i-1)+1,4*(i-1)+2,4*(i-1)+3,4*(i-1)+4
     end do
     close (60)
  end if
  close(io_bad)
  return
end subroutine check_areas
```