```
MODULE power_method

  USE iso_fortran_env, ONLY: WP => REAL64
  IMPLICIT NONE

CONTAINS

  SUBROUTINE power_li(a,x,tol,iter,lambda,clave)
    ! ------------------------------------------------------------
    ! Calculo del autovalor dominante y su respectivo autovector
    ! por el método de la potencia (versión norma infinito)
    ! ------------------------------------------------------------
    ! Argumentos de la subrutina:
    ! ------------------------------------------------------------
    ! Matriz del problema
    REAL(WP), INTENT(IN) :: a(:,:)
    !
    ! Vector inicial (entrada) / Estimación del autovector (salida)
    REAL(WP), INTENT(INOUT) :: x(:)
    !
    ! Tolerancia prefijada para el criterio de paro
    REAL(WP), INTENT(IN) :: tol
    !
    ! Máximo de iteraciones (entrada) /iteraciones realizadas (salida)
    INTEGER , INTENT(INOUT) :: iter
    !
    ! Estimación del autovalor dominante
    REAL(WP), INTENT(OUT) :: lambda
    !
    ! Clave de error: 0 ok, 1 iteraciones máximas alcanzado
    INTEGER, INTENT(OUT) :: clave
    ! ------------------------------------------------------------
    ! Variables locales
    ! ------------------------------------------------------------
    INTEGER                 :: i
    INTEGER                 :: p(1)
    REAL(WP)                :: lambda0
    REAL(WP), ALLOCATABLE   :: y(:)
    ! ------------------------------------------------------------
    ! Procedimiento
    ! ------------------------------------------------------------
    clave   = 1
    lambda0 = 0.0_WP
    ALLOCATE(y(SIZE(x)))
    p  = MAXLOC(ABS(x))
    x  = x/x(p(1))
    DO i = 1, iter
       y     = MATMUL(A,x)
       lambda= y(p(1))
       p     = MAXLOC(ABS(y))
       x     = y/y(p(1))
       IF ( ABS(lambda-lambda0) <= tol*ABS(lambda) ) THEN
          clave = 0
          iter  = i
          EXIT
       ENDIF
       lambda0 = lambda
    ENDDO
  END SUBROUTINE power_li

END MODULE power_method
```