program xingyulin

implicit none

integer, parameter :: n = 100

real, allocatable :: A(:,:)

real, allocatable :: x(:)

real, allocatable :: b(:)

real, allocatable :: r(:)

real, allocatable :: p(:)

real, allocatable :: Ap(:)

real :: alpha, beta, rr, rr\_old

integer :: i, iter\_max

real, parameter :: wucha = 1.0e-5

logical :: converged

call chushihua A

call chushihua b

x = 0.0

r = b

p = r

iter\_max = 10000

converged = .false.

do i = 1, iter\_max

call lele(A, x, Ax)

alpha = dot\_product(r, r)/dot\_product(p, Ap)

x = x + alpha \* p

r = r - alpha \* Ap

rr = dot\_product(r, r)

if (sqrt(rr) < wucha) then

converged = .true.

exit

endif

beta = rr / rr\_old

p = r + beta \* p

rr\_old = rr

end do

if (converged) then

print \*, "PCG converged in ", i, " iterations."

else

print \*, "PCG did not converge within ", iter\_max, " iterations."

endif

subroutine chushihua A

real, intent(out) :: A(:,:)

integer :: i

do i = 1, n

A(i, i) = 4.0

if (i > 1) A(i, i-1) = -1.0

if (i < n) A(i, i+1) = -1.0

end do

end subroutine chushihua A

subroutine chushihua b

real, intent(out) :: b(:)

integer :: i

do i = 1, n

b(i) = 1.0

end do

end subroutine chushihua b

subroutine lele(A, x, Ax)

real, intent(in) :: A(:,:)

real, intent(in) :: x(:)

real, intent(out) :: Ax(:)

integer :: i, j

do i = 1, n

Ax(i) = 0.0

do j = 1, n

Ax(i) = Ax(i) + A(i, j) \* x(j)

end do

end do

end subroutine lele

end program xingyulin