```
subroutine bzero(A,u,m,J)
  implicit none
  ! computes the jacobian F’(U)
  ! stores cholesky factorisation components in J
  integer, intent(in) :: m
  real(kind=8), intent(in), dimension(m,(m-1)**2) :: A
  real(kind=8), intent(in), dimension((m-1)**2)::u
  real(kind=8), intent(out), dimension(m,(m-1)**2)::J
  integer :: i,info
  real(kind = 8):: y,beta,lambda
  beta = 0.12_8
  lambda = 0.19_8
  J=A
  ! computes F’(U) = A - diag(G’(U))
  do i=1,(m-1)**2
     y = 1.0_8 + beta*u(i)
     !J banded so diag is mth row
     J(m,i) = J(m,i) - lambda*exp(u(i)/y)/(y*y)
  end do
  ! computes the cholesky factorisation and saves in J
  call dpbtrf(’U’,(m-1)**2,m-1,J,m,info)
end subroutine bzero
```