

```
subroutine func(A,u,r,beta,lambda,n_loc)

  use header
  implicit none
  ! Evaluates the nonlinear function  $F(U) = AU - G(U)$ 
  include "mpif.h"
  type(matrix), intent(in) :: A
  type(vector), intent(in) :: u
  type(vector), intent(inout) :: r
  type(vector) :: b
  integer :: n_loc
  real(kind=8), intent(in) :: beta, lambda

  allocate(b%xx(u%n))
  b%ibeg = u%ibeg
  b%iend = u%iend
  b%n = u%n

  ! Total number of unknowns
  ! Compute the nonlinear part  $G(U)$  first. This is done by copying and scaling the distributed vector to the distributed r
  call dcopy(n_loc,exp(u%xx(u%ibeg:u%iend)/(1.0_8+beta*u%xx(u%ibeg:u%iend))),1,r%xx(r%ibeg:eg),1)
  ! This scales the vector by lambda
  call dscal(n_loc,lambda,r%xx(r%ibeg),1)
  ! Compute AU
  call Mat_Mult(A,u,b)
  ! Compute  $AU - G(U)$  and stores the answer in r
  ! compute  $r = -r$  initially using dscal for efficiency
  call dscal(n_loc,-1.0_8,r%xx(r%ibeg),1)
  call daxpy(n_loc,1.0_8,b%xx(b%ibeg),1,r%xx(r%ibeg),1)
  ! deallocate vector b
  deallocate(b%xx)
end subroutine func
```