```
!PRINT DATE > Mon Mar 13 01:34:13 PM KST 2023
MODULE MD_FILTER
IMPLICIT NONE

CONTAINS
!BAND PASS FILTER
SUBROUTINE bpf_butter(f_real, f_imag, ROWS, f_l, f_h, k, filter3, bpf_real, bpf_imag
)
IMPLICIT NONE
REAL*8, DIMENSION(ROWS),INTENT(IN) :: f_real
REAL*8, DIMENSION(ROWS),INTENT(IN) :: f_imag
INTEGER, INTENT(IN) :: ROWS
INTEGER, INTENT(IN) ::  f_h, f_l
INTEGER, INTENT(IN) ::  k

INTEGER :: I, X, BS, BE, f
REAL :: T
REAL*8, PARAMETER :: pi = Acos(-1.0)

REAL*8, DIMENSION(ROWS) :: filter       !FILTER
REAL*8, DIMENSION(ROWS) :: filter2      !FILTER
REAL*8, DIMENSION(ROWS) :: filter3      !FILTER
REAL*8, DIMENSION(ROWS) :: bpf_real     !f_real after band-pass filter
REAL*8, DIMENSION(ROWS) :: bpf_imag     !f_imag after band-pass filter

!FMT = "(5F20.10)"

!READ(10,FMT) FFT
!PRINT *, k, f_c

!==========BUTTERWORTH FILTER==========
!DO  f = 1, ROWS
!    filter(f) = SQRT(1.0 - (1.0/ (1.0 + ( (REAL(f)/REAL(f_h))**(2.0 * k) ) ))) &
!     *SQRT(1.0-( ((REAL(f)/REAL(f_l))**(2.0*k))/(1+(REAL(f)/REAL(f_l))**(2.0*k))))
!END DO

DO  f = 1, ROWS
    filter(f) = SQRT(1.0/ (1.0 + ( (REAL(f)/REAL(f_h))**(2.0 * k) ) )) &
     *SQRT( ((REAL(f)/REAL(f_l))**(2.0*k))/(1+(REAL(f)/REAL(f_l))**(2.0*k)))
END DO


!FLIP THE FILTER
DO f = 1, ROWS
   filter2(f) = filter(ROWS-f+1)
END DO

DO f = 1, ROWS
   filter3(f) = filter(f) + filter2(f)
END DO
!====================================


DO f = 1, ROWS
   !PRINT*, filter(f)
   bpf_real(f) =  filter3(f) * f_real(f)
   bpf_imag(f) =  filter3(f) * f_imag(f)
   !PRINT*, bpf_real(f), bpf_imag(f)

END DO

END SUBROUTINE


!HIGT PASS FILTER
SUBROUTINE hpf_butter(f_real, f_imag, ROWS, f_c, k, filter3, hpf_real, hpf_imag)
IMPLICIT NONE
REAL*8, DIMENSION(ROWS),INTENT(IN) :: f_real
REAL*8, DIMENSION(ROWS),INTENT(IN) :: f_imag
INTEGER, INTENT(IN) :: ROWS
INTEGER, INTENT(IN) ::  f_c
INTEGER, INTENT(IN) ::  k
```

```
INTEGER :: I, X, BS, BE, f
REAL :: T
REAL*8, PARAMETER :: pi = Acos(-1.0)

REAL*8, DIMENSION(ROWS) :: filter       !FILTER
REAL*8, DIMENSION(ROWS) :: filter2      !FILTER
REAL*8, DIMENSION(ROWS) :: filter3      !FILTER
REAL*8, DIMENSION(ROWS) :: hpf_real     !f_real after high-pass filter
REAL*8, DIMENSION(ROWS) :: hpf_imag     !f_imag after high-pass filter

!FMT = "(5F20.10)"

!READ(10,FMT) FFT
!PRINT *, k, f_c

!==========BUTTERWORTH FILTER===========
DO  f = 1, ROWS
    filter(f) = SQRT(1.0-(1.0/ (1.0 + ( (REAL(f)/REAL(f_c))**(2.0 * k) ) ) ))
END DO

!FLIP THE FILTER
DO f = 1, ROWS
   filter2(f) = filter(ROWS-f+1)
END DO

DO f = 1, ROWS
   filter3(f) = filter(f) + filter2(f) - 1.0
!   filter3(f) = filter(f) + filter2(f)
END DO
!====================================

DO f = 1, ROWS
   hpf_real(f) =  filter3(f) * f_real(f)
   hpf_imag(f) =  filter3(f) * f_imag(f)
END DO

END SUBROUTINE



!HIGT PASS FILTER
SUBROUTINE hpf(f_real, f_imag, ROWS, BAND, TW, hpf_real, hpf_imag)
IMPLICIT NONE
REAL*8, DIMENSION(ROWS),INTENT(IN) :: f_real
REAL*8, DIMENSION(ROWS),INTENT(IN) :: f_imag
INTEGER, INTENT(IN) :: ROWS
INTEGER, INTENT(IN) ::  BAND

INTEGER :: I, X, BS, BE
REAL :: T
REAL*8, PARAMETER :: pi = Acos(-1.0)

REAL*8, DIMENSION(ROWS) :: TW           !TIME WINDOW
REAL*8, DIMENSION(ROWS) :: hpf_real     !f_real after high-pass filter
REAL*8, DIMENSION(ROWS) :: hpf_imag     !f_imag after high-pass filter

!FMT = "(5F20.10)"

!READ(10,FMT) FFT

!==========HIGH-PASS FILTER===========
BS = BAND                               !BAND START
BE = (ROWS-BAND)+1                      !BAND END
T = 1

DO X = 1, BS
!    TW(X) = EXP(-(((X-1)-(BAND * T))/ (REAL(BAND)/2.0)*T)**2)
    TW(X) = EXP(-(((X-1)-(BAND * T))/ 2*T)**2)
END DO

DO X = BS+1, ROWS-BAND
    TW(X) = 1.0
END DO
```

```
DO X = BE, ROWS
!   TW(X) = EXP(-(((ROWS-X)-(BAND * T))/ (REAL(BAND)/2.0)*T)**2)
   TW(X) = EXP(-(((ROWS-X)-(BAND * T))/ 2*T)**2)
END DO
!=====================================


DO X = 1, BS
   hpf_real(X) =  TW(X) * f_real(BAND+1)
   hpf_imag(X) =  TW(X) * f_imag(BAND+1)
END DO


DO X = BS+1,ROWS-BAND
   hpf_real(X) =  TW(X) * f_real(X)
   hpf_imag(X) =  TW(X) * f_imag(X)
END DO

DO X = BE, ROWS
   hpf_real(X) =  TW(X) * f_real(ROWS-BAND)
   hpf_imag(X) =  TW(X) * f_imag(ROWS-BAND)
END DO


END SUBROUTINE

END MODULE
```