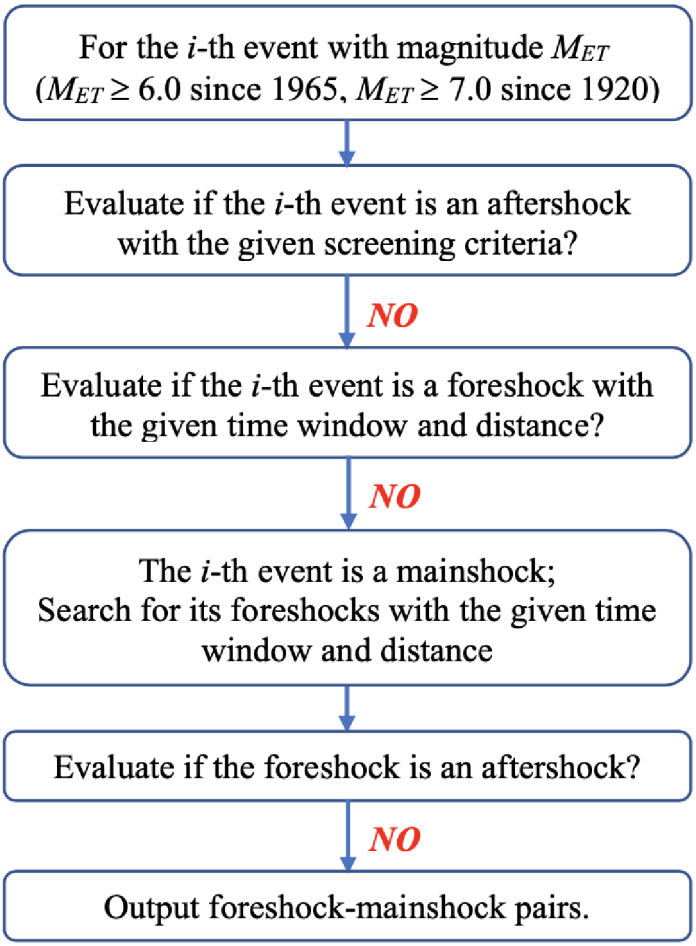

Flow diagram of the searching procedure for the M6.0+ foreshock-mainshock pairs.

```
  program search_foreshocks
  parameter (nn=101000,pi=3.141592653,RR=6371.0)

  character*50 tim(nn),inpfile,oupfile1,oupfile2,nam(nn)
  real(kind=4) :: lat(nn),lon(nn), dep(nn),mag(nn),mag1,mag2
  real(kind=8) :: dat(nn)
  integer :: ie1(2000,10)
  xlen=2.0
  dtr=pi/180.0
  ie1=0
  ! dd1 = aftershock search time window in days;
  ! spatial range for aftershock is given by radius R=xlen x 10^(-2.44+0.59*mag)+20 km
  dd1=30
  print *, "dist range, time window, and min magntiude for the aftershock (E1) and
mainshock (E2): "
  read(*,*) dist0,dd0,mag1,mag2
  read(*,*) inpfile,oupfile1,oupfile2
  read(*,*) nevt
  open(5,file=inpfile)
  open(65,file=oupfile1)
  open(66,file=oupfile2)
  open(67,file='Aftershocks')

  do i=1,nevt
     read(5,*) tim(i),lat(i),lon(i),dep(i),mag(i),dat(i),nam(i)
  enddo

  ii=0
  ii1=0
  ka=0
  do i=1,nevt   ! i = index for the mainshock
     if(mag(i).ge.mag2.and.(    (mag(i).lt.6.95 .and. dat(i).ge.23741.0) &
                           .or.(mag(i).ge.6.95 .and. dat(i).ge.7304.0 ) ) ) then
     x1=lat(i)*dtr
     y1=lon(i)*dtr
 !! Evaluate if the i-th event is a foreshock or an aftershock. If yes, stop

! Evaluate if it is an aftershock
! The spatial range of aftershock is given by the radius R = xlen x
10^(-2.44+0.59*mag) + 20 km
     k2=i
     jj=i-1
     do while((dat(i)-dat(jj)).le.dd1.and.jj.ge.1)
         x2=lat(jj)*dtr
         y2=lon(jj)*dtr
         dist=sqrt((acos(sin(x1)*sin(x2)+cos(x1)*cos(x2)*cos(y1-y2))*RR)**2+(dep(jj)-
dep(i))**2)
         dist1=xlen*10**(-2.44+0.59*mag(jj))+20.0
         if(mag(jj).gt.mag(k2).and.dist.le.dist1) k2=jj
         jj=jj-1
     enddo

! If yes (k2<i), stop searching, i->i+1
     if(k2.lt.i) then
        ka=ka+1
        write(67,'(i4," ",A24,2f8.3,2f7.1,f16.8,"  ",A20,A24,2f8.3,2f7.1,f16.8)') ka,
&

tim(i),lat(i),lon(i),dep(i),mag(i),dat(i),nam(i),tim(k2),lat(k2),lon(k2),dep(k2),mag(k2),dat(k2)
     endif

! If it is not an aftershock (k2=i); then evaluate if it is a foreshock?
```

```
! If it is a foreshore == Any event larger in dd0 (days) within the radius of dist0
(km)?
     if(k2.eq.i) then
     k1=i
     jj=i+1
 ! If yes (k1>i), it is a foreshock; stop searching, i->i+1
     do while((dat(jj)-dat(i)).le.dd0.and.jj.le.nevt)
         x2=lat(jj)*dtr
         y2=lon(jj)*dtr
         dist=sqrt((acos(sin(x1)*sin(x2)+cos(x1)*cos(x2)*cos(y1-y2))*RR)**2+(dep(jj)-
dep(i))**2)
         if(dist.le.dist0.and.mag(jj).ge.mag(i)) k1=jj
         jj=jj+1
     enddo

!! Next, search for its foreshocks
     if(k1.eq.i) then
        j=i-1          ! j = index for foreshocks
        k=0
        ddt=dat(i)-dat(j)
        do while(ddt.le.dd0.and.ddt.ge.5.8e-5)
           x2=lat(j)*dtr
           y2=lon(j)*dtr
           dist=sqrt((acos(sin(x1)*sin(x2)+cos(x1)*cos(x2)*cos(y1-y2))*RR)**2+(dep(j)-
dep(i))**2)
           if(dist.le.dist0.and.mag(j).le.mag(i)) then ! foreshock can have same
magnitude as the mainshock
 ! evaluate if this potential foreshock is an aftershock of other mainshock?
              jj=j-1
              k3=j
              do while((dat(j)-dat(jj)).le.dd1.and.jj.ge.1)
                 x3=lat(jj)*dtr
                 y3=lon(jj)*dtr
                 dist=sqrt((acos(sin(x3)*sin(x2)+cos(x3)*cos(x2)*cos(y3-y2))*RR)**2+
(dep(jj)-dep(j))**2)
                 dist1=xlen*10**(-2.44+0.59*mag(jj))+20.0
                 if(dist.le.dist1.and.mag(jj).gt.mag(j)) k3=jj
                 jj=jj-1
              enddo
! if it is not an aftershock, count it as the foreshock we want
              if(k3.eq.j) then
                 k=k+1
! k - index of foreshock for each mainshock:
! k=1 if only one foreshock and
! k=2 if two foreshocks with gradually increasing magnitudes (otherwise the 2nd EQ is
just aftershock of the 1st EQ).
                 kk=j
                 ii1=ii1+1
                 if(k.eq.1) ii=ii+1
! ii - index of mainshocks with foreshock(s), ii=1, 2, 3, ...
                    x1=lat(i)*dtr
                    y1=lon(i)*dtr
                    x2=lat(j)*dtr
                    y2=lon(j)*dtr
                    dist=sqrt((acos(sin(x1)*sin(x2)+cos(x1)*cos(x2)*cos(y1-
y2))*RR)**2+(dep(j)-dep(i))**2)
                    ddt=dat(i)-dat(j)
                 write(65,'(i4,i4,"  ",A24,2f8.3,2f7.1,"
",A24,2f8.3,2f7.1,f8.1,f11.6,f16.8,"   ",A30)' ) &

ii,k,tim(i),lon(i),lat(i),dep(i),mag(i),tim(j),lon(j),lat(j),dep(j),mag(j),dist,ddt,dat(i),nam(i)
                 write(*,'(i4,i4,"  ",A24,2f8.3,2f7.1,"
```

```
",A24,2f8.3,2f7.1,f8.1,f11.6,f16.8,"   ",A30)' ) &

ii,k,tim(i),lon(i),lat(i),dep(i),mag(i),tim(j),lon(j),lat(j),dep(j),mag(j),dist,ddt,dat(i),nam(i)
                  ie1(ii,k)=j
               endif
            endif
            j=j-1
            ddt=dat(i)-dat(j)
         enddo
      endif
      endif
      endif
  enddo
  close(65)
  close(5)
  ne1=ii

  do i=1,ne1
     do j=1,10
     k1=0
     do k=j+1,10
        if(ie1(i,k).ge.1) then
           l=ie1(i,j)
           m=ie1(i,k)
           if(mag(l).ge.mag(m)) then
              ii=ii+1
              k1=k1+1
              ddt=dat(l)-dat(m)
              x1=lat(l)*dtr
              y1=lon(l)*dtr
              x2=lat(m)*dtr
              y2=lon(m)*dtr
              dist=sqrt((acos(sin(x1)*sin(x2)+cos(x1)*cos(x2)*cos(y1-y2))*RR)**2+
(dep(j)-dep(i))**2)
              if(dist.le.dist0) then
                  write(66,'(i4,i4,"  ",A24,2f8.3,2f7.1,"
",A24,2f8.3,2f7.1,f8.1,f11.6,f16.8,"   ",A30)' ) &

ii,k1,tim(l),lon(l),lat(l),dep(l),mag(l),tim(m),lon(m),lat(m),dep(m),mag(m),dist,ddt,dat(l),nam(l)
              endif
           endif
        endif
     enddo
     enddo
  enddo
  close(66)
  close(67)
  end
```