Appendix D. Code for Dynamic Kinocilum Simulation

```
      Block data bldata
C-----------------------------------------------------------------
C     def(iy)  deflection at node i
C     vel(iy)  velocity at node i
C     xload(iy) load/EI at node i
C     uf(iy)   fluid velocity at node i
C     dt       time step
C     dy       geometric mesh size
C     Cdrag    Coefficient of drag
C     diam     diameter of kinocilia
C     vel0     initial velocity of fluid
C     pltc     plate time constant
C     n        number of nodes in kinocilium
C     nf       number of fluid nodes
C
C-----------------------------------------------------------------
$INCLUDE params.h
      common /d1/def(Mxnode)
      common /v1/vel(Mxnode)
      common /b1/bend(Mxnode)
      common /u1/uf(Mxfnode)
C
      common /rundata/dt, dy, n, nf, time
      common /kindata/asp
      common /fludata/visckin, vel0,  pltc
C
      double precision def
      double precision vel
      double precision bend
      double precision uf
C
      real dt, dy
      integer n, nf
      double precision time
      double precision asp
      real visckin, vel0,  pltc
C
      end
      Program dyn_kin
C-----------------------------------------------------------------------
C
C     Program unit: models a kinocilia dynamically in a flow
C --- applied BC
C     Description:
C     Coded: 3/14/96
C     Modified Last:
C
C     yloop    logical (T/F) if done looping
C     ptime    time to print results
C
$INCLUDE params.h
C
      logical yloop
      double precision ptime
C
```

```
      data  yloop/.true./
C
C     open files for output
C
      call openfl
C
C     load the parameters and initial conditions
C
      call loaddat
      call loadinit
C
C     set counter and call print routine to get initial printout
C
      icnt = 0
      call prtres(ptime, icnt)
C
C     loop over each timestep
C
      do while (yloop)
         icnt = icnt + 1
         call solvexp
         call solveflu
         call prtres(ptime, icnt)
         call stpcond(yloop, n)
      enddo
C
      end
      Subroutine loaddat
C-----------------------------------------------------------------------
C
C     Program unit: loads all scalars
C     Description:
C     Coded: 3/14/96
C     Modified Last:
C
C     dt       time step
C     dy       geometric mesh size
C     Cdrag    Coefficient of drag
C     diam     diameter of kinocilia
C     vel0     initial velocity of fluid
C     n        number of nodes
C-----------------------------------------------------------------
$INCLUDE params.h
C
      common /rundata/dt, dy, n, nf, time
      common /kindata/asp
      common /fludata/visckin, vel0,  pltc
C
      real dt, dy
      integer n, nf
      double precision time
      double precision asp
      real vel0, Cdrag, visckin, pltc
C
      read(2,*) dt, dy
      read(2,*) n, nf
      read(2, *) asp
      read(2,*) visckin
      read(2,*) vel0
      read(2,*) pltc
```

```
      read (2,*)dummy
C
      if (n .lt. 10) then
         write (*,*) 'Error: n must be greater than 7'
         stop
      endif
C
      return
      end
      Subroutine loadinit
C-----------------------------------------------------------------------
C
C     Program unit: loads initial condition of velocity and deflection
C     Description:
C     Coded: 3/14/96
C     Modified Last:
C
C     def(iy)  deflection at node i
C     vel(iy)  velocity at node i
C     load(iy) load/EI at node i
C     dy       geometric mesh size
C     Cdrag    Coefficient of drag
C     diam     diameter of kinocilia
C     vel0     initial velocity of fluid
C     n        number of nodes
C-----------------------------------------------------------------
C
$INCLUDE params.h
C
      common /d1/def(Mxnode)
      common /v1/vel(Mxnode)
      common /b1/bend(Mxnode)
      common /u1/uf(Mxfnode)
C
      common /rundata/dt, dy, n, nf, time
C
      real dt, dy
      double precision time
      integer n, nf
C
C --- load kinocilium initial conditions
C
      do iy = 1, n
         vel(iy) = 0.
         def(iy) = 0.
         bend(iy) = 0.
      enddo
C
C --- load fluid initial condition
C
      do iy = 1, nf
         uf(iy) = 0.
      enddo
      time = 0.
C
      return
      end
      subroutine openfl
C-----------------------------------------------------------------------
C
```

```
C     Program unit:
C     Description:
C     Coded:
C     Modified Last:
C
      open (2, file = 'dk.in')
      open (10, file = 'tipdef.out')
      open (11, file = 'tipvel.out')
      open (12, file = 'defpro.out')
      open (13, file = 'velpro.out')
      open (14, file = 'fvelpro.out')
      return
      end
      Subroutine prtres(ptime, icnt)
C-----------------------------------------------------------------------
C
C     Program unit:
C     Description:
C     Coded: 3/14/96
C     Modified Last:
C
$INCLUDE params.h
      common /d1/def(Mxnode)
      common /v1/vel(Mxnode)
      common /b1/bend(Mxnode)
      common /u1/uf(Mxfnode)
C
      common /rundata/dt, dy, n, nf, time
C
      double precision def
      double precision vel
      double precision bend
      double precision uf
C
      real dt, dy
      double precision time
      integer n, nf
C
      double precision ptime
C
      if (ptime .le. time) then
C
C     tip deflection
C
      write(*,'(e12.5, 3d15.6)') time, def(n), vel(n), uf(n)
         write (10,*) time, def(n)
C
C     tip velocity
C
         write (11,*) time, vel(n)
C
C     deflection profile
C
         write (12, '(50d12.3)') time, (def(iy), iy=n,1, -1)
C
C     velocity profile
C
         write (13, '(50d12.3)') time, (vel(iy), iy=n,1, -1)
         write (14, '(200d12.3)') time, (uf(iy), iy=nf,1, -1)
         call stdata
```

```
c         write (*,*) 'Enter next time for reporting...'
         read (2,*)  ptime, dummy
         if (dummy .gt. 0) dt = dummy
      endif
C
C
      return
      end
      Subroutine rldata
C-----------------------------------------------------------------
C     def(iy)  deflection at node i
C     vel(iy)  velocity at node i
C     xload(iy) load/EI at node i
C     uf(iy)   fluid velocity at node i
C     dt       time step
C     dy       geometric mesh size
C     Cdrag    Coefficient of drag
C     diam     diameter of kinocilia
C     vel0     initial velocity of fluid
C     pltc     plate time constant
C     n        number of nodes in kinocilium
C     nf       number of fluid nodes
C
C-----------------------------------------------------------------
$INCLUDE params.h
      common /d1/def(Mxnode)
      common /v1/vel(Mxnode)
      common /b1/bend(Mxnode)
      common /u1/uf(Mxfnode)
C
      common /rundata/dt, dy, n, nf, time
      common /kindata/asp
      common /fludata/visckin, vel0,  pltc
C
      double precision def
      double precision vel
      double precision bend
      double precision uf
C
      real dt, dy
      integer n, nf
      double precision time
      double precision asp
      real vel0, visckin, pltc
C
      open(15, file = 'store.dat')
      read(15,*) dt, dy, n, nf, time
      read(15,*) asp
      read(15,*) visckin, vel0,  pltc
      read(15,*) (def(i), i=1,n)
      read(15,*) (vel(i), i=1,n)
      read(15,*) (bend(i), i=1,n)
      read(15,*) (uf(i), i=1,nf)
      close(15)
C
      end
      Program dyn_kin
C-----------------------------------------------------------------------
C
C     Program unit: models a kinocilia dynamically in a flow
```

```
C --- applied BC
C     Description:
C     Coded: 3/14/96
C     Modified Last:
C
C     yloop    logical (T/F) if done looping
C     ptime    time to print results
C
$INCLUDE params.h
C
      logical yloop
      double precision ptime
C
      data  yloop/.true./
C
C     open files for output
C
      call openfl
C
C     load the parameters and initial conditions
C
      call rldata
C
C     set counter and call print routine to get initial printout
C
      icnt = 0
      ptime = 0
C
C     loop over each timestep
C
      do while (yloop)
         icnt = icnt + 1
         call solvexp
         call solveflu
         call prtres(ptime, icnt)
         call stpcond(yloop, n)
      enddo
C
      end
      subroutine solveflu
C-----------------------------------------------------------------------
C
C     Program unit:
C     Description:
C     Coded: 3/14/96
C     Modified Last:
C
$INCLUDE params.h
C
      common /u1/uf(Mxfnode)
C
      common /rundata/dt, dy, n, nf, time
      common /fludata/visckin, vel0,  pltc
C
      double precision uf
      double precision u(Mxfnode)
C
      real dt, dy
      integer n, nf
      double precision time
```

```
      real visckin, vel0, pltc
C
      double precision coeff
C
      if (pltc .lt. 0.) then
         do i=1, nf
            u(i) = float(i)/float(nf)*vel0
         enddo
      else
         coeff = dt/dy/dy*visckin
         u(1) = 0
         do i=2, nf - 1
            u(i) = coeff*uf(i+1) - (2.*coeff - 1.)*uf(i) + coeff*uf(i-1)
         enddo
         u(nf) = vel0*exp(-float(time)/pltc)
      endif
C
      time = time + dt
      do i = 1, nf
         uf(i) = u(i)
      enddo
C
      return
      end
      Subroutine solvexp
C-----------------------------------------------------------------------
C
C     Program unit: solves for load ( (d^4 (def))/ dy^4)
C     Description: uses finite difference equations for the fourth
C                  derivative of deflection w.r.t. y to dy^2 accuracy.
C     Coded: 3/14/96
C     Modified Last:
C
C     def(iy)  deflection at node i
C     load(iy) load/EI at node i
C     dy       geometric mesh size
C     n        number of nodes
C
C-----------------------------------------------------------------------
C
$INCLUDE params.h
      common /d1/def(Mxnode)
      common /v1/vel(Mxnode)
      common /b1/bend(Mxnode)
      common /u1/uf(Mxfnode)
C
      common /rundata/dt, dy, n, nf, time
      common /kindata/asp
      common /fludata/visckin, vel0,  pltc
C
      double precision def
      double precision vel
      double precision bend
      double precision uf
      double precision v(Mxnode)
      double precision b(Mxnode)
C
      real dt, dy
      integer n, nf
      double precision time
```

```
      double precision asp
      real viscin, vel0,  pltc
C
      double precision dtdy, pi, coeff, utild, rey, cdrag
      pi = 3.141592
C
C --- centered difference
C
      dtdy = dt/dy/dy
      coeff = 2.*dt/pi
      vchg = 0.
      bchg = 0.
C
      do iy = 1, n
         utild = uf(iy)-vel(iy)
         rey = 2*dabs(utild)/viscin/(asp*asp)
         if (rey .gt. 0) then
            cdrag = 8*Pi/rey/(dlog(8/rey) - 0.077216)
         else
            cdrag = 0
         endif
         if (iy .eq. 1) then
            b(1) = 2*(vel(2) - vel(1))*dtdy
            v(1) = 0.
         else if (iy .lt. n) then
            b(iy) = (vel(iy+1) - 2*vel(iy) + vel(iy-1))*dtdy
            v(iy) = -(bend(iy+1) - 2*bend(iy) + bend(iy-1))*dtdy +
     1              coeff*cdrag*dabs(utild)*utild
         else if (iy .eq. n) then
            b(n) = 0.
            v(n) = 2.*(bend(n) - bend(n-1))*dtdy +
     1             coeff*cdrag*dabs(utild)*utild
         endif
      enddo
C
C --- integrate velocity for deflection and update primary variables
C
      do iy = 1, n
         def(iy) = def(iy) + dt/2.*(v(iy) + 2*vel(iy))
         vel(iy)= v(iy) + vel(iy)
         bend(iy) = b(iy) + bend(iy)
      enddo
C      if (time .gt. 3) write (*,'(e12.7, 3d20.12)')
C     1    time, vel(n), vel(iy) - uf(iy),v(n)
C
      return
      end
      Subroutine stdata
C-----------------------------------------------------------------
C     def(iy)  deflection at node i
C     vel(iy)  velocity at node i
C     xload(iy) load/EI at node i
C     uf(iy)   fluid velocity at node i
C     dt       time step
C     dy       geometric mesh size
C     Cdrag    Coefficient of drag
C     diam     diameter of kinocilia
C     vel0     initial velocity of fluid
C     pltc     plate time constant
C     n        number of nodes in kinocilium
```

```
C     nf       number of fluid nodes
C
C-----------------------------------------------------------------
$INCLUDE params.h
      common /d1/def(Mxnode)
      common /v1/vel(Mxnode)
      common /b1/bend(Mxnode)
      common /u1/uf(Mxfnode)
C
      common /rundata/dt, dy, n, nf, time
      common /kindata/asp
      common /fludata/visckin, vel0,  pltc
C
      double precision def
      double precision vel
      double precision bend
      double precision uf
C
      real dt, dy
      integer n, nf
      double precision time
      double precision asp
      real vel0, visckin, pltc
C
      open(15, file = 'store.dat')
      write(15,*) dt, dy, n, nf, time
      write(15,*) asp
      write(15,*) visckin, vel0,  pltc
      write(15,*) (def(i), i=1,n)
      write(15,*) (vel(i), i=1,n)
      write(15,*) (bend(i), i=1,n)
      write(15,*) (uf(i), i=1,nf)
      close(15)
C
      end
      Subroutine stpcond(yloop)
C-------------------------------------------------------------------------
C
C     Program unit:
C     Description:
C     Coded: 3/14/96
C     Modified Last:
C
$INCLUDE params.h
      common /v1/vel(Mxnode)
      common /rundata/dt, dy, n, nf, time
C
      double precision vel
C
      real dt, dy
      integer n, nf
      double precision time
C
      logical yloop
C
c      if (time .gt. 10) then
c         dt = 1e-1
c      else if (time .gt. 1) then
c         dt = 1e-4
c      else if (time .gt. .1) then
```

```
c         dt = 1e-3
c      else if (time .gt. .01) then
c         dt = 1e-5
c      endif
C
c      yloop = abs(vel(n)) .gt. 1e-5 .or.
c     1        time .lt. 2
      yloop = time .lt. 20000
C
      return
      end
```