Supplementary material:

Laminar-Turbulent transition in channel flow with superhydrophobic surfaces modelled as a partial slip wall

F. Picella, J.-C. Robinet, S. Cherubini

1 Subroutines for partial slip (Robin) boundary condition in Nek5000

In this supplementary material we describe the modifications needed to implement a partial slip (Robin) boundary condition (BC) within Nek5000[2].

The idea is to convert the native *symmetry* BC, which is nothing else than a Neumann condition as follows:

$$\partial u/\partial y = 0, \quad v = 0, \quad \partial w/\partial y = 0 \tag{1}$$

into a *Robin* one, namely, a weighted Neumann-Dirichlet combination:

$$u + L_s \partial u/\partial y = 0, \quad v = 0, \quad w + L_s \partial w/\partial y = 0, \tag{2}$$

with no mass flow through the boundary, here considered to be normal to the y direction. As shown in the literature [3], the influence of a superhydrophobic surface over an overlying flow can be taken into account by accurately tuning the scalar quantity L_s, to which we refer as *slip length*[1].

A Robin boundary condition is already implemented in Nek5000 for scalar variables (temperature or other passive scalars). As suggested by some developers[1], this BC can be reverse-engineered so to obtain a Navier BC for the velocity field. Looking at the present implementation for scalar fields in subroutine CDSCAL(), the Robin BC is obtained by adding on top of the "H2" matrix a diagonal matrix "S", upon call of BCNEUSC(). In our framework, the counterpart of CDSCAL() is the PLAN3() subroutine, where the boundary conditions are imposed to the velocity field. The latter is the only subroutine that has been modified and is reported in the listing (1). The elements added to the original "H2" matrix are computed using an additional subroutine, BCNEUSC_MOD(), reported in listing (2) and derived from BCNEUSC().

In order to minimize the changes to the original code, we have decided to modify an existing primitive BC rather than defining a new one. We have chosen to modify the primitive

[1] https://lists.mcs.anl.gov/pipermail/nek5000-users/2017-April/004287.html

’SYM’ BC (see the symmetry condition in equation (1)) because of its inherent resemblance with the desired Robin one in equation (2). Thus, all element faces in the *.rea/*.re2 mesh files denoted with the three letter char variable ’SYM’, will be subject to the Robin equation (2).
The scalar slip length L_s must be provided by the user within the *.rea file, as well as the flag ’IFSTRS’ has to be set to ’false’. In the present implementation, the single scalar value for L_s is provided to the solver using one of the blank parameter variables left by the developers to the users, here chosen arbitrarily to be parameter(79). Obviously, for $L_s = 0.0$, a no-slip BC is obtained, while for large slip lengths, the symmetry BC is retrieved.

In the following listing, comments to the code are indicated as

```
!ROBIN_BC
```

1.1 Source file modification

Robin BC implementation is made by modifying only the ../TRUNK/NEK/PLANX.F file.

```
      SUBROUTINE PLAN3 (IGEOM)
C-----------------------------------------------------------------------
C
C     Compute pressure and velocity using consistent approximation spaces.
C     Operator splitting technique.
C
C-----------------------------------------------------------------------
      include 'SIZE'
      include 'INPUT'
      include 'EIGEN'
      include 'SOLN'
      include 'TSTEP'
      include 'GEOM'!ROBIN_BC
      include 'MASS'!ROBIN_BC
C
      COMMON /SCRNS/  RESV1 (LX1,LY1,LZ1,LELV)
     $ ,              RESV2 (LX1,LY1,LZ1,LELV)
     $ ,              RESV3 (LX1,LY1,LZ1,LELV)
     $ ,              DV1   (LX1,LY1,LZ1,LELV)
     $ ,              DV2   (LX1,LY1,LZ1,LELV)
     $ ,              DV3   (LX1,LY1,LZ1,LELV)
      COMMON /SCRVH/  H1    (LX1,LY1,LZ1,LELV)
     $ ,              H2    (LX1,LY1,LZ1,LELV)
      COMMON /MYROB/  H2_ROB(LX1,LY1,LZ1,LELV) !ROBIN_BC
      n=nx1*ny1*nz1*nelv !ROBIN_BC
C
      IF (IGEOM.EQ.1) THEN
C
C        Old geometry
C
         CALL MAKEF
```

```
C
      ELSE
C
C        New geometry, new b.c.
C
         INTYPE = -1
         CALL SETHLM  (H1,H2,INTYPE)
C-----------------------------------------------------------------!
C--------Create H2 that takes into account for Robin BC           !
         call rzero(H2_ROB,N) !ROBIN_BC                           !
         call bcneusc_mod(H2_ROB)!ROBIN_BC                        !
C--------Add contribution of Robin BC to standard Neumann condition!
         call add2(H2, H2_ROB,N) !ROBIN_BC                        !
C-----------------------------------------------------------------!
         CALL CRESVIF (RESV1,RESV2,RESV3,H1,H2)
         mstep = abs(param(94))
         if (param(94).ne.0. .and. istep.ge.mstep) then
          call ophinv_pr(dv1,dv2,dv3,resv1,resv2,resv3,h1,h2,tolhv,nmxh)
c         CALL OPHINV  (DV1,DV2,DV3,RESV1,RESV2,RESV3,H1,H2,TOLHV,NMXH)
         else
       CALL OPHINV  (DV1,DV2,DV3,RESV1,RESV2,RESV3,H1,H2,TOLHV,NMXH)
C            call ophinv_rob (DV1,DV2,DV3,RESV1,RESV2,RESV3,
C     &                        H1,H2,H2_ROB,TOLHV,NMXH)
         endif
         CALL OPADD2  (VX,VY,VZ,DV1,DV2,DV3)
c
c        Default Filtering
c
c        alpha_filt = 0.05
c        if (param(103).ne.0.) alpha_filt=param(103)
c        call q_filter(alpha_filt)
c
c        CALL SSNORMD (DV1,DV2,DV3)
c
         call incomprn(vx,vy,vz,pr)
C
      ENDIF
C
      RETURN
      END
C
C*******************************************************************
```

Listing 1: modified plan3 subroutine

```
C*******************************************************************
      SUBROUTINE BCNEUSC_MOD(S) !PICELLA
      include 'SIZE'
      include 'TOTAL'
      include 'CTIMER'
      include 'NEKUSE'
      DIMENSION S(LX1,LY1,LZ1,LELV)
```

```fortran
      CHARACTER CB*3
      real NU
        slip_length=param(79)+1e-10! To be imposed within the .REA !
        NU = abs(param(02))
      if(nio.eq.0)write(6,*)'SLIP_LENGTH',SLIP_LENGTH,'NU',NU
      NFACES=2*NDIM
      NXYZ  =NX1*NY1*NZ1
      IFIELD=1 !flow variables
      NEL   =NELFLD(IFIELD)
      NTOT  =NXYZ*NEL
C      CALL RZERO(S,NTOT)
C      if(nio.eq.0)
C     $write(6,*)'SUBROUTINE␣BCNEUSC_MOD'
C     $          ,NFACES,NEL,NTOT,LELV
         DO 1000 IE=1,NEL
         DO 1000 IFACE=1,NFACES
            ieg=lglel(ie)
            CB =CBC(IFACE,IE,IFIELD)
            IF (CB.EQ.'SYM') THEN !Apply the condition to 'SYM' faces only...
               IA=0
C
C IA is areal counter, assumes advancing fastest index first. (IX...IY...IZ)
C
               CALL FACIND (KX1,KX2,KY1,KY2,KZ1,KZ2,NX1,NY1,NZ1,IFACE)
               DO 100 IZ=KZ1,KZ2
               DO 100 IY=KY1,KY2
               DO 100 IX=KX1,KX2
                  IA = IA + 1
!                 S(IX,IY,IZ,IE)=1/slip_length
                  S(IX,IY,IZ,IE) = S(IX,IY,IZ,IE) +
     $            NU/Slip_length*AREA(IA,1,IFACE,IE)/BM1(IX,IY,IZ,IE)
C                 if(nio.eq.0)write(6,*)IE,S(IX,IY,IZ,IE)
 100           CONTINUE
            endif
 1000    CONTINUE
      RETURN
      END
C
```

Listing 2: subroutine bcneusc_mod

As indicated within the subroutine, the slip length L_s value is prescribed within the *.REA file, using the blank variable PARAMETER(79).

1.2 Anisotropic Robin BC

While anisotropic slip length has not been employed in the present work, it can be easily implemented by computing multiple H2 matrices, one for each spatial direction. They can be be formed separately using different BCNEUSC_MOD() like subroutines, obtaining two H2_X and H2_Z matrices, each related to a given slip length. At this point, anisotropic slip length for the velocity components can be achieved using a sligthly modified version of the OPHINV() subroutine, where the following section:

```
[...]
            CALL HMHOLTZ ('VELX',OUT1,INP1,H1,H2,V1MASK,VMULT,
     $                                IMESH,TOLH,NMXI,1)
            CALL HMHOLTZ ('VELY',OUT2,INP2,H1,H2,V2MASK,VMULT,
     $                                IMESH,TOLH,NMXI,2)
            IF (NDIM.EQ.3)
     $      CALL HMHOLTZ ('VELZ',OUT3,INP3,H1,H2,V3MASK,VMULT,
     $                                IMESH,TOLH,NMXI,3)
[...]
```

Listing 3: standard ophinv subroutine, written within the NAVIER4.F file

should be replaced by the following one:

```
[...]
            CALL HMHOLTZ ('VELX',OUT1,INP1,H1,H2_X,V1MASK,VMULT,!ROBIN_BC
     $                                IMESH,TOLH,NMXI,1)
            CALL HMHOLTZ ('VELY',OUT2,INP2,H1,H2,V2MASK,VMULT,
     $                                IMESH,TOLH,NMXI,2)
            IF (NDIM.EQ.3)
     $      CALL HMHOLTZ ('VELZ',OUT3,INP3,H1,H2_Z,V3MASK,VMULT,!ROBIN_BC
     $                                IMESH,TOLH,NMXI,3)
[...]
```

Listing 4: ophinv_anisotropic subroutine

References

[1] Taegee Min and John Kim. "Effects of hydrophobic surface on stability and transition". In: *Physics of Fluids* 17.10 (2005), p. 108106. DOI: 10.1063/1.2126569. eprint: https://aip.scitation.org/doi/pdf/10.1063/1.2126569. URL: https://aip.scitation.org/doi/abs/10.1063/1.2126569.

[2] James W. Lottes Paul F. Fischer and Stefan G. Kerkemeier. *nek5000 Web page*. http://nek5000.mcs.anl.gov. 2008.

[3] Jonathan P. Rothstein. "Slip on Superhydrophobic Surfaces". In: *Annual Review of Fluid Mechanics* 42.1 (2010), pp. 89–109. DOI: 10.1146/annurev-fluid-121108-145558. eprint: https://doi.org/10.1146/annurev-fluid-121108-145558. URL: https://doi.org/10.1146/annurev-fluid-121108-145558.