Computation of the Wall Shear Stress Distribution on Unstructured Grids

Mehdi Najafi

BioMedical Simulation Lab, Department of Mechanical & Industrial Engineering, University of Toronto

April 7, 2020

Contents

1 Theory from Continuum Mechanics

Let $\vec{u}(\vec{x}, t)$ be a continuously differentiatable velocity vector field describing the fluid flow at a point $\vec{x}$ in the domain of interest $\Omega \subset \mathbb{R}^3$. Using indicial notation for the components of $\vec{u}$, the components of strain rate tensor, $\bar{\bar{\dot{\varepsilon}}}$, are defined by

$$\dot{\varepsilon}_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \text{ for } i, j = 1, 2, 3 \tag{1}$$

where $u_{i,j} = \partial u_i / \partial x_j$. Let $\vec{n}$ be the inward facing unit vector normal to the wall. The Cauchy formula gives the overall viscous stress on the wall for an incompressible Newtonian fluid:

$$\vec{T}_n = 2\mu\,\bar{\bar{\dot{\varepsilon}}} \cdot \vec{n} \tag{2}$$

where μ is the dynamic viscosity of the fluid. The wall overall viscous stress, $\vec{T}_n$, also known as the traction vector, is a vector that has wall normal and tangential contributions:

$$\vec{T}_n = \vec{\tau}_n + \vec{\tau}_w \tag{3}$$

The normal contribution can be found as

$$\vec{\tau}_n = (\vec{T}_n \cdot \vec{n})\vec{n} \tag{4}$$

The tangential contribution or the wall shear stress can be computed as the difference between the wall overall viscous stress and its projection onto the wall normal direction:

$$\vec{\tau}_w = \vec{T}_n - (\vec{T}_n \cdot \vec{n})\vec{n} \tag{5}$$

Now, substituting Eq. (2) into Eq. (5) gives

$$\vec{\tau}_w = 2\mu\,(\bar{\bar{\dot{\varepsilon}}} \cdot \vec{n} - (\bar{\bar{\dot{\varepsilon}}} \cdot \vec{n} \cdot \vec{n})\vec{n}) \tag{6}$$

and in indicial notation

$$\vec{\tau}_{wi} = 2\mu\,(\dot{\varepsilon}_{ij} n_j - (\dot{\varepsilon}_{kj} n_j n_k) n_i) \tag{7}$$

Substituting the strain rate tensor, it gives the wall shear stress in terms of velocity gradients and surface normal vectors on the wall

$$\vec{\tau}_{wi} = \mu\,((u_{i,j} + u_{j,i}) n_j - (u_{k,j} + u_{j,k}) n_j n_k n_i) \tag{8}$$

Also, the wall shear rate is

$$\vec{\sigma}_{wi} = (u_{i,j} + u_{j,i}) n_j - (u_{k,j} + u_{j,k}) n_j n_k n_i \tag{9}$$

2 Numerical Computation

As described in the previous section, having the velocity gradients and wall normal vectors one can compute the wall shear stress. Numerical approximation of wall shear rate is feasible using Vascular Modeling Toolkit (VMTK) package [1] using *vmtkmeshshearrate* script that minimizes a quadrature based formula on adjacent points for each grid point. Unfortunately, this script only gives $u_{i,j} n_j$ as wall shear rate and ignores $(u_{j,i} n_j - (u_{k,j} + u_{j,k}) n_j n_k n_i)$. The implementation of the C++ code behind *vmtkmeshshearrate* script is given in the code listing 1.

Listing 1: VMTK implementation of wall shear rate out of velocity gradients since July 27, 2006 - vtkVmtk/Misc/vtkvmtkMeshWallShearRate.cxx

```
  for (i=0; i<numberOfPoints; i++)
    {
    velocityGradientArray->GetTuple(i,velocityGradient);
    normalsArray->GetTuple(i,normal);
    for (j=0; j<3; j++)
      {
      wallShearRate[j] = -normal[0] * velocityGradient[3*j + 0] - normal[1] * velocityGradient[3*j + 1] - normal[2] * velocityGradient[3*j + 2];
      }
    wallShearRateArray->SetTuple(i,wallShearRate);
    }
```

I made a fix in that VMTK routine and re-built the resulting package which is labeled as ‘*new*’ while the original one is labeled as ‘*old*’ in what comes later in this report. The new implementation in the code behind *vmtkmeshshearrate* script is changed as in the code listing 2.

Listing 2: Fixed VMTK implementation of wall shear stress out of velocity gradients - vtkVmtk/Misc/vtkvmtkMeshWallShearRate.cxx

```
  double nSn, Sn[3], strainRateTensor[3][3];
  for (i=0; i<numberOfPoints; i++)
  {
    velocityGradientArray->GetTuple(i,velocityGradient);
    normalsArray->GetTuple(i,normal);

    // compute the strain rate tensor
    strainRateTensor[0][0] = 0.5*(velocityGradient[0] + velocityGradient[0]);
    strainRateTensor[0][1] = 0.5*(velocityGradient[1] + velocityGradient[3]);
    strainRateTensor[0][2] = 0.5*(velocityGradient[2] + velocityGradient[6]);

    strainRateTensor[1][0] = 0.5*(velocityGradient[3] + velocityGradient[1]);
    strainRateTensor[1][1] = 0.5*(velocityGradient[4] + velocityGradient[4]);
    strainRateTensor[1][2] = 0.5*(velocityGradient[5] + velocityGradient[7]);

    strainRateTensor[2][0] = 0.5*(velocityGradient[6] + velocityGradient[2]);
    strainRateTensor[2][1] = 0.5*(velocityGradient[7] + velocityGradient[5]);
    strainRateTensor[2][2] = 0.5*(velocityGradient[8] + velocityGradient[8]);

    // determine the projection of strain rate tensor on normal direction: StrainRate.normal and normal.StrainRate.normal
    nSn = 0.0;
    for (j=0; j<3; j++)
    {
      Sn[j] = strainRateTensor[j][0]*normal[0] + strainRateTensor[j][1]*normal[1] + strainRateTensor[j][2]*normal[2];
      nSn += normal[j] * Sn[j];
    }

    // normal vectors are outward facing, so negative of 2*(Sn[j] - nSn * normal[j]) should be the correct wall shear rate components
    for (j=0; j<3; j++)
    {
      wallShearRate[j] =  2.0 * ( nSn * normal[j] - Sn[j] );
    }
    wallShearRateArray->SetTuple(i,wallShearRate);
  }
```

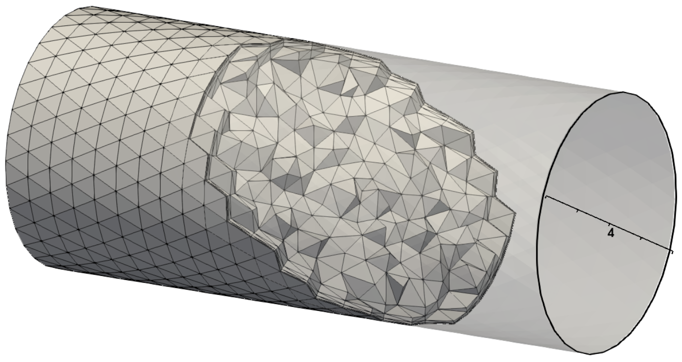

Figure 1: A cylinder of diameter 4, stacked on the x-y plane.

3 Benchmark: flow inside a circular cylinder

As described in the previous section, having the velocity gradients and wall normal vectors one can compute the wall shear stress. As the benchmark, let's look at the flow inside a halo circular cylinder with radius of 2 and length of $L = 9.41$, stacked on x-y plane. The surface normal vectors are in the opposite direction of rays off the center at each cross-section. The cylinder volume is filled with tetrahedrons using the VMTK mesh generator which is depicted in Fig.1. In order to determine the exact value of wall shear stress using Eq.(8) the velocity gradients as well as the wall normal vectors are required. Since the cylinder is stacked on x-y plane or aligned with z-axis, the normal vectors at the wall have no z-component.

$$\vec{n} = (\frac{-x}{\sqrt{x^2+y^2}}, \frac{-y}{\sqrt{x^2+y^2}}, 0) \tag{10}$$

3.1 Parabolic Axial Flow

Consider a first order parabolic flow in z-direction:

$$\vec{u} = (0, 0, 1 - \frac{r^2}{4}) \tag{11}$$

From (8), the correct value of wall shear stress is constant 1. In Fig. 2 the flow field, the numerical values of wall shear stress using old and new VMTK script is shown. It is important to note that the numerical approximation of

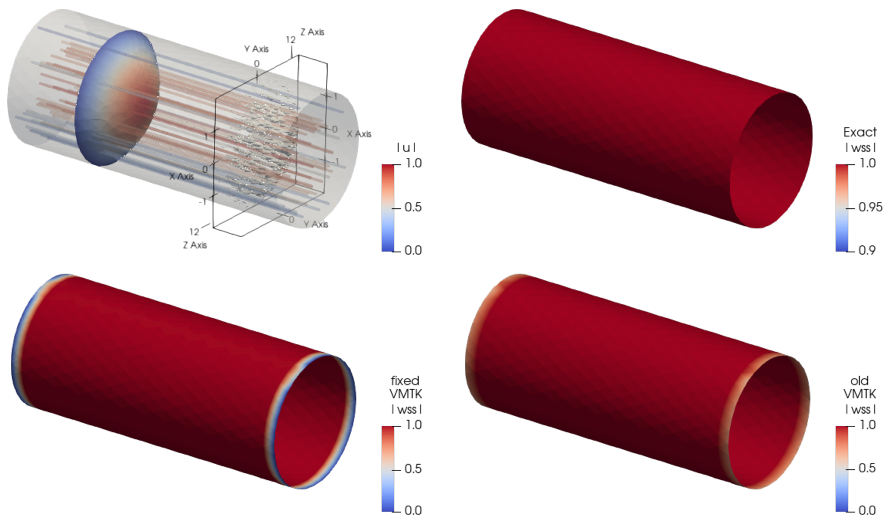

Figure 2: Uniform parabolic velocity field and the corresponding wall shear stress distribution.

gradients in VMTK for the points on the intersection of non-similar boundaries is the same as internal ones, which makes a non-negligible error in those regions for any gradient based computations like wall shear stress here. Therefore, the error in small round regions at the inlet/outlet is related to the way gradients are approximated and not related to the rest of computations.

3.2 Periodic Transversal Flow

Now, consider a periodic flow in x-direction:

$$\vec{u} = (sin(\frac{4\pi}{L}z), 0, 0) \tag{12}$$

which has only one nonzero derivative of

$$u_{1,3} = \frac{\partial u_3}{\partial z} = \frac{4\pi}{L}cos(\frac{4\pi}{L}z) \tag{13}$$

in the strain rate tensor. Taking surface normal vectors from Eq. (10), the exact value of wall shear stress using (8) is

$$\vec{\tau}_w = (0, 0, \frac{-x}{\sqrt{x^2+y^2}}\frac{4\pi}{L}cos(\frac{4\pi}{L}z)) \tag{14}$$

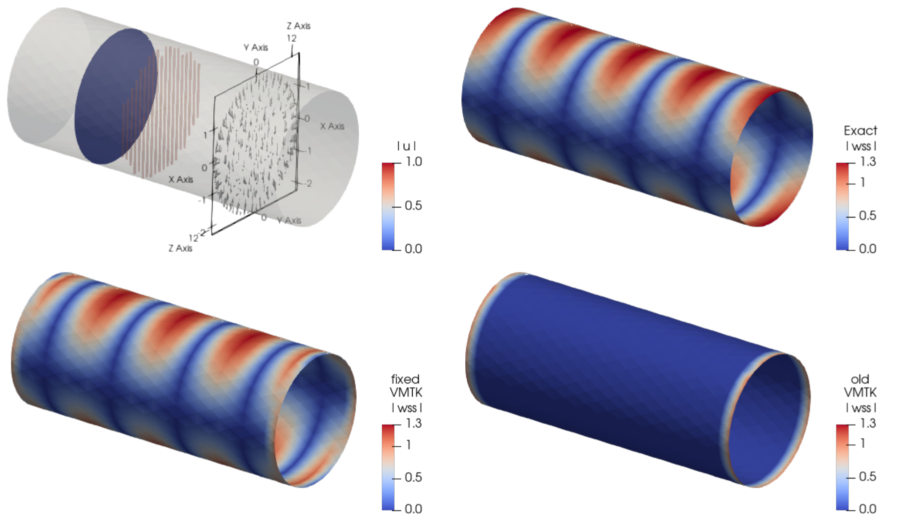

Figure 3: Periodic transversal velocity field and the corresponding wall shear stress distribution.

Figure 3 shows the resulting velocity field, exact values and numerical approximation of wall shear stress from the Eq. (12). Here, the discrepancy between old and new VMTK codes are obviously because of those ignored terms mentioned before.

3.3 Mixed Parabolic Axial and Periodic Transversal Flow

The final benchmark is the combination of previous velocity fields, i.e.,

$$\vec{u} = (sin(\frac{4\pi}{L}z), 0, 1 - \frac{r^2}{4}) \tag{15}$$

As the strain rate and the wall shear rate are linearly related to the velocity gradients, the wall shear stress in this case is simply the linear combination of previous ones, i.e.,

$$\vec{\tau}_w = (0, 0, 1 + \frac{-x}{\sqrt{x^2+y^2}} \frac{4\pi}{L} cos(\frac{4\pi}{L}z)) \tag{16}$$

Figure 4 shows the resulting velocity field, exact values and numerical approximation of wall shear stress from the Eq. (15). Here, the old VMTK wall shear rate script only revealed the axial flow contributions while the new VMTK code gave the correct one.

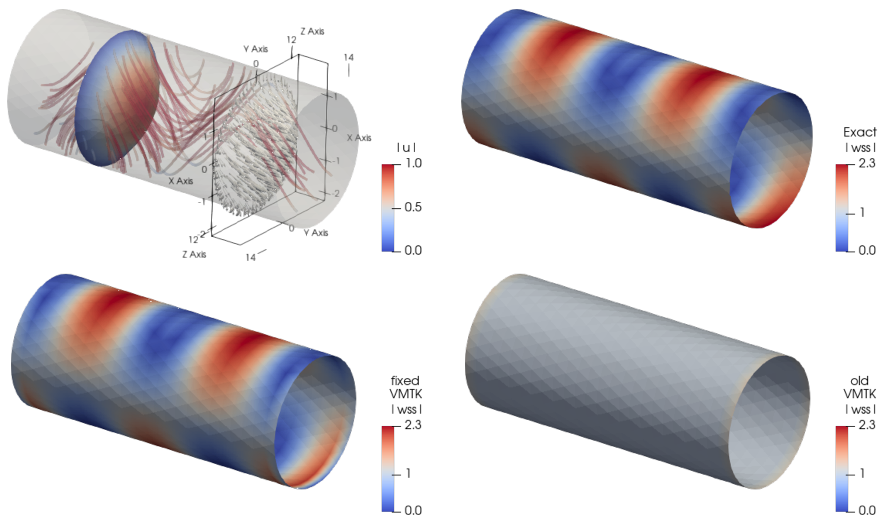

Figure 4: Mixed uniform parabolic (in z direction) and periodic transversal (in x direction) velocity field and the corresponding wall shear stress distribution.

References

[1] Antiga L, Piccinelli M, Botti L, Ene-Iordache B, Remuzzi A and Steinman DA. An image-based modeling framework for patient-specific computational hemodynamics. *Medical and Biological Engineering and Computing*, 46: 1097-1112, Nov 2008.