OpenFOAM®
Journal

Volume 3 [Technical Notes], Pages 159–176
ISSN: 2753-8168

Verification and Validation of the Spalart Allmaras Model with Rotation and Curvature Correction for Incompressible and Compressbile Flows

M. Alletto*
Email address: `michael.alletto@gmx.de`

DOI: https://doi.org/10.51560/ofj.v3.97
Results with version(s): OpenFOAM®v2112
Repository: https://gitlab.com/mAlletto/openfoamtutorials/-/tree/master/SpalartAllmarasRCsend

Abstract. The Spalart-Allmaras turbulence model is the model which gets the highest technological readiness level in the NASA turbulence model resource home page. The model is widely used to predict external incompressible and compressbile flows. Different corrections to this model incorporating additional physical effects are presented on the same home page. Unfortunately none of these corrections are available in the official OpenFOAM version. Furthermore, no publicly available repository which contains the full set of corrections suggested by Shur et al. [1] to account for curvature and rotation effects was found. In this technical note, the rotation and streamline curvature correction is incorporated into the Spalart-Allmaras model in order to overcome this limitation. The same equations as suggested in Shur et al. [1] are implemented and verified by means of the incompressible flow in a rotating channel, a 2D bump flow in a channel and a channel flow exhibiting a U-turn. Regarding the rotating channel flow and the flow in a U-turn some quantitative differences remain with respect to the results of Shur et al. [1]. For the rotating channel flow, a careful consistency check with respect to the analytical relations leading to the computation of the factor multiplied with the production term was made. No inconsistencies in the implementation where found. Unfortunately the exact reason for the differences between the present results compared to the computations of Shur et al. [1], was not found. The interested readers are encouraged to help to clarify these issues. In the fourth test case it is shown that the model predicts a more correct turbulent viscosity distribution in the transonic flow around a delta wing. Applying the streamline and curvature corrections to the Spalart-Allmaras model predicts an earlier vortex breakdown compared to the standard Spalart-Allmaras model for a high angle of attack configuration. The agreement with the reference experiments is much better when using the corrections proposed by Shur et al. [1] to the Spalart-Allmaras compared to the standard Splart-Allmaras model.

1. Introduction

Streamline curvature and the rotation of the frame of reference have an astonishing effect on turbulence even if mildly present. The direct numerical simulation (DNS) of Moser and Moin [2] studied the influence of streamline curvature on turbulence by means of a slightly curved periodic channel. The authors found a visible effect on the symmetry of the velocity fluctuations and the shear stress even for a channel curvature of 79δ (δ is the channel half width). Moser and Moin [2] observed that the velocity fluctuations and the shear stress are smaller on the convex (inner) half of the channel compared to the concave (outer) half of the channel. The influence of the rotation of the frame of reference on turbulence is similar. Brethouwer et al. [3] performed a parameter study on the influence of the rotation rate Ω of the frame of reference on the turbulent flow through a channel with periodic boundary conditions. The higher the rotation rate the bigger the influence on turbulence. For a sufficiently high rotation rate Ω the velocity profile obtained even a parabolic shape which leads to the conclusion that the flow was almost laminar. As discussed in [3, 4] the parameter influencing whether turbulence is augmented or diminished is $S = -\frac{2\Omega}{dU/dy}$. For $S < -1$ and $S > 0$ turbulence is damped and for $-1 < S < 0$ turbulence is augmented.

In order to account for these effects, Spalart and Shur [5] and Shur et al. [1] introduced a factor f_{r1} which is multiplied with the production term in the Spalart-Allmaras model. The authors demonstrated the positive effect of the correction on the prediction of flows subject to streamline curvature and rotation. The full set of equation can be found in Shur et al. [1]. Even if Shur et al. [1] proposed their correction more than two decades ago, unfortunately no publicly available implementation in OpenFOAM of the

* Corresponding author

Received: 9 December 2022, Accepted: 22 September 2023, Published: 5 November 2023

correction described in Shur et al. [1] could be found. A few implementations of curvature corrections within OpenFOAM could be found in the literature. Alahmadi et al. [6], e.g., incorporated a curvature correction and rotation correction into Menter's SST model. The authors found that, if they accounted for the streamline curvature and the rotation, they could fairly well reproduce the solid-body-like tangential velocity distribution present in the center of a cyclone separator. A solid-body-like tangential velocity distribution is zero in the center of the cyclone separator and increases linearly with the radius. Using the standard SST model leads to a prediction of the maximum of the tangential velocity in the center of the cyclone. Such a velocity distribution is typical for a potential flow vortex and not for the one of a solid-body-like vortex. Another example of incorporating a rotation and curvature correction into an OpenFOAM code can be found in [7]. The authors accessed the performance of a curvature correction incorporated into an explicit algebraic Reynolds stress model. The authors found that accounting for streamline curvature and rotation effects increases the accuracy of the results in a centrifugal pump flow when compared to PIV data. Another work to mention is the one of You et al. [8]. The authors studied the ability of the SST model of Menter together with a streamline curvature correction (SSTRC) to predict the flow and heat transfer in a swirl tube. You et al. [8] found that the performance of the SSTRC model is similar to the more computationally demanding DES model.

Summing up, there are a few works where a streamline and curvature correction is implemented in turbulence models available in OpenFOAM. Unfortunately however, no publicly available repository was found which contains all terms proposed by Shur et al. [1] and also suggested on the turbulence model homepage of the NASA. In order to remedy to the deficiency of no publicly available Spalart-Allmaras model with the streamline and curvature correction proposed by Shur et al. [1], the model was implemented, verified and validated by means of four different test cases. The source code and the test cases together with the scripts used to generate the plots are attached to the case files provided.

2. **Description of the implementation**

Before starting with the description of the implementation, we will shorty recall the standard Spalart-Allmaras model. After that, we will see how the corrections for the streamline curvature and the rotation are included. The standard Spalart-Allmaras model implemented in the OpenFOAM version used for the current investigation reads:

$$\frac{D}{Dt}(\rho\tilde{\nu}) = \nabla\cdot(\rho D_{\tilde{\nu}}\tilde{\nu}) + \frac{C_{b2}}{\sigma_{\nu_t}}\rho|\nabla\tilde{\nu}|^2 + C_{b1}\rho\tilde{S}\tilde{\nu}(1-f_{t2}) - \left(C_{w1}f_w - \frac{C_{b1}}{\kappa^2}f_{t2}\right)\rho\frac{\tilde{\nu}^2}{\tilde{d}^2} + S_{\tilde{\nu}} \tag{1}$$

Note that the f_{t2} term in the above equation is not implemented in the OpenFOAM version used. In order to account for the streamline curvature and rotation, the production term is multiplied by a factor f_{r1} (see Eqn. (4) for the definition). This means that if the factor f_{r1} is negative, turbulence is destroyed and if the factor is greater than one the production of the turbulence is higher compared to the standard model. The final model equation for $\tilde{\nu}$ reads:

$$\frac{D}{Dt}(\rho\tilde{\nu}) = \nabla\cdot(\rho D_{\tilde{\nu}}\tilde{\nu}) + \frac{C_{b2}}{\sigma_{\nu_t}}\rho|\nabla\tilde{\nu}|^2 + f_{r1}C_{b1}\rho\tilde{S}\tilde{\nu} - C_{w1}f_w\rho\frac{\tilde{\nu}^2}{\tilde{d}^2} + S_{\tilde{\nu}} \tag{2}$$

The factor f_w is calculated as follows:

$$f_w = g\left[\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right]^{1/6}, \quad g = r + c_{w2}(r^6 - r), \quad r = min\left[\frac{\tilde{\nu}}{\tilde{S}\kappa^2\tilde{d}^2}, 10\right] \tag{3}$$

The constants used throughout the work are set to following value: $\sigma_{\nu_t} = 0.666667$, $\kappa = 0.41$, $C_{b2} = 0.622$, $C_{b1} = 0.1355$, $C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1+C_{b2}}{\sigma_{\nu_t}}$, $c_{w2} = 0.622$ and $c_{w3} = 2$. $\tilde{d}$ stands for the wall distance. $S_{\tilde{\nu}}$ represent implicit or explicit source terms which can be added via the fvOptions present in OpenFOAM.

Equations (4) to (9) represent the correction described by Shur et al. [1] to account for the turbulence enhancement or attenuation caused by the streamline curvature or the rotation of the reference frame. The model was originally proposed by Spalart and Shur [5]. This reference however contains some typographical error but nevertheless some useful physical insights about the influence of system rotation and streamline curvature on turbulence can be found.

$$f_{r1} = (1+c_{r1})\frac{2r^*}{1+r^*}\left[1 - c_{r3}\tan^{-1}(c_{r2}\hat{r})\right] - c_{r1} \tag{4}$$

$$r^* = \frac{S}{\omega} \tag{5}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right),\ \omega_{ij} = \frac{1}{2}\left[\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) + 2\epsilon_{mji}\Omega_m\right] \tag{6}$$

$$S^2 = 2S_{ij}S_{ij},\ \omega^2 = 2\omega_{ij}\omega_{ij},\ D^2 = \frac{1}{2}\left(S^2 + \omega^2\right) \tag{7}$$

$$c_{r1} = 1,\ c_{r2} = 12,\ c_{r3} = 1 \tag{8}$$

$$\hat{r} = \frac{2\omega_{ik}S_{jk}}{D^4}\left[\frac{DS_{ij}}{Dt} + \left(\epsilon_{imn}S_{jn} + \epsilon_{jmn}S_{in}\right)\Omega_m\right],\ \frac{DS_{ij}}{Dt} = \frac{\partial S_{ij}}{\partial t} + u_k\frac{\partial S_{ij}}{\partial x_k} \tag{9}$$

Note that the velocity u_i and the derivatives in Eqn. (4) to Eqn. (9) are defined in the rotating frame of reference. This is an important point not mentioned in the NASA turbulence model resource home page but stated in the paper by Shur et al. [1]. The following two listings show the implementation of the rotation and curvature correction. The first task to achieve is to retrieve the rotation of the reference frame Ω. How it is done is shown in the following listing:

```
template<class BasicTurbulenceModel>
volVectorField SpalartAllmarasRC<BasicTurbulenceModel>::OmegaRefFrame() const
{
    volVectorField Omega
    (
        IOobject(
             "Omega",
              this->runTime_.timeName(),
              this->mesh_,
              IOobject::NO_READ,
              IOobject::NO_WRITE
        ),
        this->mesh_,
        dimensionedVector("Omega", dimless/dimTime, Zero)
    );

    const auto* MRFZones = this->mesh().template
                           cfindObject<IOMRFZoneList>("MRFProperties");

    if (!MRFZones && MRFZones->MRFZoneList::size() == 0 && debug)
    {
        Info << "Unable to find MRFProperties in the database. " << endl
             << "Omega set to zero" << endl;
        return Omega;
    }

    for (label i=0; i < MRFZones->MRFZoneList::size(); i++)
    {
        const dictionary& MRFZoneDict =
            MRFZones->IOdictionary::subDict(MRFZones->MRFZoneList::operator[](i).
                name());

        const label cellZoneIndex =
            this->mesh_.cellZones().findIndex(MRFZoneDict.get<word>("cellZone"));

        const labelList& cells = this->mesh_.cellZones()[cellZoneIndex];

        forAll(cells, cellI)
        {
              label celli = cells[cellI];
              Omega[celli] = MRFZones->MRFZoneList::operator[](i).Omega();
        }
     }

    if (this->mesh_.time().writeTime())
    {
        Info << "write OmegaRefFrame" << endl;
        Omega.write();
    }


    return Omega;
```

The first step performed in the above listing, is to create the volVectorField Omega. The field contains for each cell of the mesh the rotation rate of the moving reference frames in which the cell is located. The second step is to loop over all existing moving reference frames and assign to the volVectorField Omega the rotation rate of the corresponding frame of reference where the cell with the index cellI is located. It is important to get the correct index of the moving reference zone among all possible cell zones present in the domain. This task is achieved by the call:

```
        const label cellZoneIndex =
            this->mesh_.cellZones().findIndex(MRFZoneDict.get<word>("cellZone"));
```

The function call retrieves the id of the MRF zone from its name.
The next listing shows the actual computation of the factor f_{r1}.

```
template<class BasicTurbulenceModel>
tmp<volScalarField> SpalartAllmarasRC<BasicTurbulenceModel>::fr1( ) const
{

    const auto* MRFZones =
    this->mesh().template
             cfindObject<IOMRFZoneList>("MRFProperties");

    volVectorField Urel (this->U_);
    volVectorField& UrelRef (Urel);
    MRFZones->MRFZoneList::makeRelative(UrelRef);
    surfaceScalarField phi(Foam::fvc::interpolate(UrelRef)&this->mesh_.Sf());

    volVectorField Omega (this->OmegaRefFrame());
    volTensorField gradU (Foam::fvc::grad(UrelRef));

    // transpose to be consistent with notation of Shur et al. 2000
    gradU = gradU.T();
    volTensorField S ("S", scalar(0.5)  * ( gradU + gradU.T()));
    volTensorField dSdt (Foam::fvc::ddt(S));
    volTensorField H ("H", *Omega);
    volTensorField W ("W", scalar(0.5)  * ( gradU - gradU.T()  + 2.0*H));
    volTensorField WS (W & S.T());
    volScalarField S2 (scalar(2.0) * (S && S));
    volScalarField W2 (scalar(2.0) * (W && W));
    volScalarField D2 (scalar(0.5) * (S2 + W2));
    dimensionedScalar smallW2 (W2.dimensions(), SMALL);
    volScalarField rstar ("rstar", sqrt(S2) / (sqrt(W2 + smallW2)));


    volTensorField ejmnSinOm ("ejmnSinOm", S & H.T());
    volTensorField eimnSjnOm ("eimnSjnOm", H & S.T());


    volTensorField DSDT (dSdt + Foam::fvc::div(phi,S));

    dimensionedScalar smallD2 (D2.dimensions()*D2.dimensions(), SMALL);

    volTensorField WSD ("WSD", scalar(2.0) * WS / (sqr(D2) + smallD2 ));
    volScalarField rhat ("rhat", WSD && (DSDT + eimnSjnOm + ejmnSinOm));

    return ( (scalar(1.0) + Cr1_)*scalar(2.0)*rstar / (scalar(1.0) + rstar)* (scalar
        (1.0) - Cr3_*Foam::atan(Cr2_*rhat)) - Cr1_ );

}
```

It is important to note here, that according to Shur et al. [1] the velocity and the derivatives are defined in the rotating frame of reference. Unfortunately this peculiarity is not mentioned in the turbulence model resource homepage. In OpenFOAM the velocity is defined in the inertial frame of reference. For this reason the velocity has to be made relative to the rotating frame of reference before that the gradient operator is applied to the velocity. This is done in the following code snippet:

```
    const auto* MRFZones =
    this->mesh().template
            cfindObject<IOMRFZoneList>("MRFProperties");

    volVectorField Urel (this->U_);
    volVectorField& UrelRef (Urel);
    MRFZones->MRFZoneList::makeRelative(UrelRef);
```

Once the velocity is made relative, the gradient operator can be used to calculated the expressions required for the model. Another point to mention here is the difference in the definition of the gradient of a vector adopted in OpenFOAM and the one used in Shur et al. [1]. In OpenFOAM the gradient of the velocity is defined as $\nabla\mathbf{u} = \frac{\partial u_j}{\partial x_i}$ while in Shur et al. [1] the gradient of the velocity is defined as $\nabla\mathbf{u} = \frac{\partial u_i}{\partial x_j}$. In order to be consistent with the notation adopted in Shur et al. [1], the following operation had to be performed in the code (i.e. we have to transpose the velocity gradient computed in OpenFOAM):

```
    gradU = gradU.T();
```

Another point worth to mention here is how the inner product **P** of two tensors **T** and **S** are defined in OpenFOAM: $P_{ij} = T_{ik}S_{kj}$. This means that the summation goes over the columns of the first tensor and the rows of the second tensor. In Eqn. (9) we need to compute the inner product $\omega_{ik}S_{jk}$, i.e. we need to do the summation over the columns of both tensors. In order to achieve the latter inner product with the operator overloaded in OpenFOAM we need to transpose the second tensor S_{jk}:

```
    volTensorField WS (W & S.T());
```

All other steps leading to the computation of the terms required for the factor f_{r1} are rather standard except for the computation of the terms $(\epsilon_{imn}S_{jn} + \epsilon_{jmn}S_{in})\,\Omega_m$ and $2\epsilon_{mji}\Omega_m$. For the computation of the three tensors $2\epsilon_{mji}\Omega_m$, $\epsilon_{imn}S_{jn}\Omega_m$ and $\epsilon_{jmn}S_{in}\Omega_m$ we first have to recall the definition of the the Hodge star operator $\star$. If we apply the operator $\star$ to the rotation rate vector Ω we get the matrix **H**: $\mathbf{H} = \star\Omega$. The matrix **H** looks like:

$$\mathbf{H} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}$$

In order to see how the operator **H** and the tensor $2\epsilon_{mji}\Omega_m$ are related we write down the tensor:

$$\epsilon_{mji}\Omega_m = \begin{bmatrix} 0 & \epsilon_{321}\Omega_3 & \epsilon_{231}\Omega_2 \\ \epsilon_{321}\Omega_3 & 0 & \epsilon_{132}\Omega_1 \\ \epsilon_{213}\Omega_2 & \epsilon_{123}\Omega_1 & 0 \end{bmatrix}$$

At this point we have to recall that the permutation symbol ϵ_{imn} which is equal to 1 for even permutations, i.e. $\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1$ and equal to -1 for odd permutations, i.e. $\epsilon_{132} = \epsilon_{312} = \epsilon_{213} = -1$. If we insert the definition in the above relation we get:

$$\epsilon_{mji}\Omega_m = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}$$

If we compare the above relation with Eqn. (2) we see that both are equal.

The next step is to write down the tensor $\epsilon_{imn}S_{jn}\Omega_m$:

$$\epsilon_{imn}S_{jn}\Omega_m = \begin{bmatrix} \epsilon_{123}S_{13}\Omega_2 + \epsilon_{132}S_{12}\Omega_3 & \epsilon_{123}S_{23}\Omega_2 + \epsilon_{132}S_{22}\Omega_3 & \epsilon_{123}S_{33}\Omega_2 + \epsilon_{132}S_{32}\Omega_3 \\ \epsilon_{231}S_{11}\Omega_3 + \epsilon_{213}S_{13}\Omega_1 & \epsilon_{231}S_{21}\Omega_3 + \epsilon_{213}S_{23}\Omega_1 & \epsilon_{231}S_{31}\Omega_3 + \epsilon_{213}S_{33}\Omega_1 \\ \epsilon_{312}S_{12}\Omega_1 + \epsilon_{321}S_{11}\Omega_2 & \epsilon_{312}S_{22}\Omega_1 + \epsilon_{321}S_{21}\Omega_2 & \epsilon_{312}S_{32}\Omega_1 + \epsilon_{321}S_{31}\Omega_2 \end{bmatrix}$$

At this point we have to recall again that the permutation symbol ϵ_{imn} which is equal to 1 for even permutations, i.e. $\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1$ and equal to -1 for odd permutations, i.e. $\epsilon_{132} = \epsilon_{312} = \epsilon_{213} = -1$. For all other combinations of indexes the permutation symbol is equal to zero. With this convention we obtain:

$$\epsilon_{imn} S_{jn} \Omega_m = \begin{bmatrix} S_{13}\Omega_2 - S_{12}\Omega_3 & S_{23}\Omega_2 - S_{22}\Omega_3 & S_{33}\Omega_2 - S_{32}\Omega_3 \\ S_{11}\Omega_3 - S_{13}\Omega_1 & S_{21}\Omega_3 - S_{23}\Omega_1 & S_{31}\Omega_3 - S_{33}\Omega_1 \\ S_{12}\Omega_1 - S_{11}\Omega_2 & S_{22}\Omega_1 - S_{21}\Omega_2 & S_{32}\Omega_1 - S_{31}\Omega_2 \end{bmatrix}$$

We can easily see that

$$\epsilon_{imn} S_{jn} \Omega_m = \mathbf{H} \cdot \mathbf{S}^T = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \cdot \begin{bmatrix} S_{11} & S_{21} & S_{31} \\ S_{12} & S_{22} & S_{32} \\ S_{13} & S_{23} & S_{33} \end{bmatrix}$$

Using the same procedure as described above, we can see that

$$\epsilon_{jmn} S_{in} \Omega_m = \begin{bmatrix} S_{13}\Omega_2 - S_{12}\Omega_3 & S_{11}\Omega_3 - S_{13}\Omega_1 & S_{12}\Omega_1 - S_{11}\Omega_2 \\ S_{23}\Omega_2 - S_{22}\Omega_3 & S_{21}\Omega_3 - S_{23}\Omega_1 & S_{22}\Omega_1 - S_{21}\Omega_2 \\ S_{33}\Omega_2 - S_{32}\Omega_3 & S_{31}\Omega_3 - S_{33}\Omega_1 & S_{32}\Omega_1 - S_{31}\Omega_2 \end{bmatrix}$$

and finally

$$\epsilon_{jmn} S_{in} \Omega_m = \mathbf{S} \cdot \mathbf{H}^T = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \cdot \begin{bmatrix} 0 & \Omega_3 & -\Omega_2 \\ -\Omega_3 & 0 & \Omega_1 \\ \Omega_2 & -\Omega_1 & 0 \end{bmatrix}$$

The next code snippets shows how the correction term f_{r1} is actually incorporated into the Spalart-Allmaras model. Note that the production term is not always positive since f_{r1} can also be negative. For this reason the source term is explicit if it is positive and implicit if it is negative.

```

    const volScalarField::Internal fr1(this->fr1());

    tmp<fvScalarMatrix> nuTildaEqn
    (
        fvm::ddt(alpha, rho, nuTilda_)
      + fvm::div(alphaRhoPhi, nuTilda_)
      - fvm::laplacian(alpha*rho*DnuTildaEff(), nuTilda_)
      - Cb2_/sigmaNut_*alpha*rho*magSqr(fvc::grad(nuTilda_))
     ==
        fvm::SuSp(fr1*Cb1_*alpha()*rho()*Stilda, nuTilda_)
      - fvm::Sp(Cw1_*alpha()*rho()*fw(Stilda)*nuTilda_()/sqr(y_), nuTilda_)
      + fvOptions(alpha, rho, nuTilda_)
    );
```

3. Test cases

In this section, the test cases used for the verification and validation of the implementation are presented. Of course all case files and the scripts used to generate the plots in this section are provided in the downloadable material. We start with the 2D bump flow in a channel. For this test case reference simulation data are available from the NASA turbulence model resource. The solver CFL3D together with the Spalart-Allmaras model with rotation and curvature correction was used to generate the data. By comparing the present implementation with the reference simulation we can check if the current implementation is correct. The second test case is the turbulent flow in a rotating channel. The purpose of this test case is to show that the influence of a rotating frame of reference on the turbulence is correctly captured by the current implementation. The third test case is the 2D flow in a U-turn. For this case we have an attenuation of turbulence close to the inner radius of the curved channel and turbulence enhancement close to the outer radius of the bend. Here we check if this behavior can be reproduced by the model. All these steps are mandatory preliminary steps to ensure the correctness of the implementation in order that one can proceed to the final test case. It is the transonic flow over the second vortex flow experiment (VFE-2) delta wing. We will see here that the introduction of the rotation and curvature correction in the Spalart-Allmaras model leads to the appearance of a shock induced vortex breakdown. The vortex breakdown is not predicted if the standard Spallart-Allmaras model is used. The

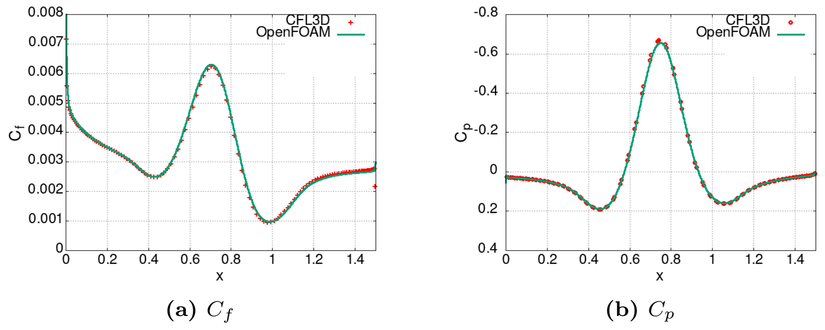


Figure 1. Friction coefficient C_f (left) and the pressure coefficient C_p (right). Comparison of the current implementation of the RC Spalart-Allmaras model (SpalartAllmarasRC) with the results of the code CFL3D with the same turbulence model.

results obtained by the Spalart-Allmaras model with rotation correction are closer to the experimental observations by Chu and Luckring [9].

3.1. **2D bump in a channel.** The first test case used for the verification of the current implementation is the 2D bump in a channel described in the NASA turbulence model resource (TMR) home page (see https://turbmodels.larc.nasa.gov/bump.html). The purpose is to check if OpenFOAM together with the Spalart-Allmaras RC model gives the same results as the reference code CFL3D. The grid is taken from the OpenFOAM tutorials (see $FOAM_TUTORIALS/tutorials/incompressible/simpleFoam/bump2D). Note that the grids provided on the TMR homepage could not be converted to an OpenFOAM readable format. In the description of the test case on the OpenFOAM homepage (see https://www.openfoam.com/documentation/guides/latest/doc/verification-validation-turbulent-bump-2d.html) the TMR homepage is cited and it is mentioned that the test case is based on the description provided therein. The grid provided in the tutorial is therefore believed to be very close to the plot3D grid provided on the TMR homepage. The grid had a size of 240×648 cells in wall-normal and streamwise directions, respectively. For the velocity a constant inflow velocity of $U_{inf} = 69.44\frac{m}{s}$ is used. At the bump a no-slip condition is used, at the outlet a inflow-outflow condition is applied and at the top a symmetry condition is imposed. For the pressure a zero gradient condition is used for the inflow and the wall, a symmetry condition for the top and a fixed value for the outlet. For the turbulent viscosity ν_t the nutLowReWallFunction wall function is used for the bump, a symmetry condition is used for the top and a calculated condition is applied for the rest of the boundaries. For $\hat{\nu}$ a fixed value of $6.93 \times 10^{-5}\frac{m^2}{s}$ is used at the inlet, an inflow-outflow condition at the outlet, a symmetry condition at the top and the value at the bump is set to zero. It was ensured that the maximum size of the cells adjacent the wall was small enough to ensure that the condition $y^+ < 1$ was satisfied. The incompressible solver simpleFoam was used to simulate this test case.

Figure 1 shows the comparison between the current implementation and the results of the code CFL3D (note that the results of the code CFL3D are computed on a grid of 641×1409). The left figure shows the friction coefficient $C_f = 0.5\frac{\tau_w}{\rho U_{inf}^2}$ and the right figure shows the pressure coefficient $C_p = 0.5\frac{p_w}{\rho U_{inf}^2}$. τ_w and p_w are the wall shear stress and the pressure at the wall, respectively. It is evident that the results of the two codes using the same turbulence model match very well. The small difference between the solvers compared may be due to the different grids used. Another reason for the difference may be that the CFL3D results are computed assuming a compressible flow while the results of OpenFOAM shown here are computed with an incompressible solver. Even though the Mach number $Ma = 0.2$ used to compute the CFL3D results was very small, small compressibility effect may still occur.

3.2. **Rotating channel.** This test case consists in a channel rotating with a constant rotational speed of $\Omega_3 = 0.5\frac{1}{s}$ around the z-axis in the counter-clockwise sense. Figure 2 shows the configuration of the test case. The flow points in the positive x-direction. The y-coordinate points from the bottom wall to the top wall. The z-direction points outward of the plane. The rotation rate Ω_3 points in the positive z-direction. The purpose of this test case is to check if the current implementation gives correct results for situations where the flow is subject to the rotation of the frame of reference. For this type of configuration we have a turbulence attenuation over the whole channel. The attenuation at the bottom wall is less pronounced compared to the top wall (see [1,3,10]). A no-slip condition is applied on the bottom and top wall and empty condition on the side patches. Inflow and outflow is periodic. The flow is forced to have a constant bulk velocity of $U_{b,i} = 0.75\frac{m}{s}$ in the streamwise direction in the inertial frame of reference. The channel

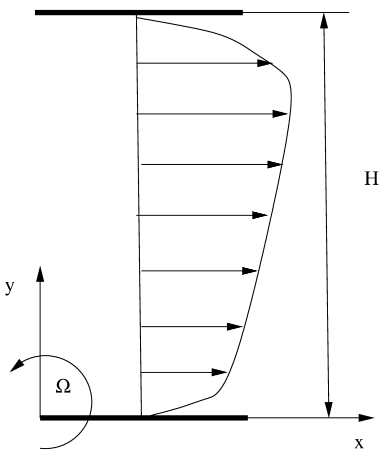
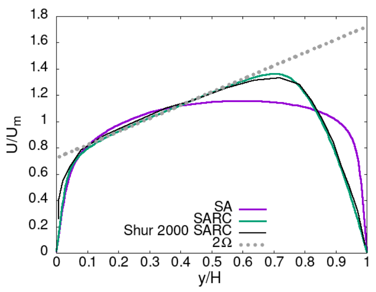


Figure 2. Sketch of the rotating channel configuration

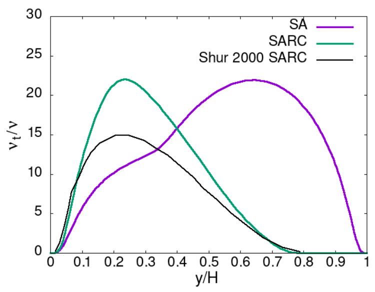
height is $H = 1m$. If we add the mean velocity of the frame of reference $U_{fr} = 0.5\Omega_3 * H = 0.25\frac{m}{s}$ to the bulk velocity in the inertial frame of reference, we obtain a mean velocity in the frame of reference of $U_m = 1\frac{m}{s}$. The Reynolds number is equal to $Re = \frac{U_m H}{\nu} = 5.8 \times 10^3$. A Rossby number of $Ro = \frac{\Omega H}{U_m} = 0.5$ is used to evaluate the implementation. For $\hat{\nu}$ a fixed value of 0 was used at the walls and for ν_t a low Re-number wall function was used. 300 cells were used in the wall-normal direction and 1 cell in the streamwise direction. It was ensured that the maximum size of the cells adjacent the wall was small enough to ensure that the condition $y^+ < 1$ was satisfied. The incompressible solver simpleFoam was used to simulate this test case.

Figure 3 compares the mean velocity scaled by the bulk velocity $\frac{U}{U_m}$ (top left) and the turbulent viscosity $\frac{\nu_t}{\nu}$ (top right) predicted by the SA model with the SARC model. The results of the SARC model shown in Shur et al. [1] (labeled as Shur 2000 SARC) are also shown for comparison. The line with a slope of 2Ω is also include for comparison. Furthermore the factor f_{r1} computed with a rotation rate of $\Omega_3 = 0.5\frac{1}{s}$ is compared to the same quantity but for a non-rotating case in the bottom center figure. It is evident that incorporating the effect of rotation in the turbulence model leads to a much more asymmetric velocity profile with respect to the channel center line compared to the model without correction. Interestingly the velocity profile predicted by the corrent implementation differ slightly from the one predicted by Shur et al. [1]. The agreement in the center of the channel with the line with a slope of 2Ω is however better compared with the results of Shur et al. [1]. Velocity profiles with a slope of 2Ω are also predicted by the DNS of Brethouwer [11] over a wide range of Reynolds numbers and for moderate Rossby numbers. For high Rossby numbers the DNS of Brethouwer [11] exhibits parabolic mean velocity profiles. The exact reason of the difference with the results of Shur et al. [1] is not very clear. A careful consistency check of the implemented equation is performed in the appendix. We can conclude with a sufficient degree of confidence from this consistency check that the implementation is correct for the current rotating channel flow. We can also observe that the value of the turbulent viscosity ν_t at the suction (upper) part is much lower compared to the pressure (lower) part of the channel for the SARC model. The opposite behavior can actually be observed when the SA model is used. The turbulence viscosity precicted by the current work is higher compared with the one computed by Shur et al. [1]. When looking at the behavior of f_{r1} we see that the factor is equal to one for the case without system rotation. The factor is positive at the lower part of the channel and negative at the upper part. Also f_{r1} differs from the one computed by Shur et al. [1]. It turns negative at a lower y-coordinate compared to the simulation of Shur et al. [1]. In the upper part of the channel a constant offset is observed with respect to the results of Shur et al. [1]. Again, the exact reason is not very clear.

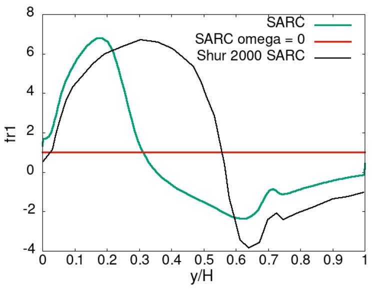
3.3. **U-Turn.** Figure 4 shows the configuration used for the present test case. The inlet is placed $50H$ from the start from the u-turn. H denotes the height of the channel. The purpose to place the inflow so far upstream was to ensure fully developed conditions at $S/H = -10$ upstream of the start of the U-turn. S represents the distance measured along the channel center line. At this position the reference experiment had fully developed conditions (see Monson et al. [12]). The vectors **n** and **t** denote the wall-normal vector and the wall tangential vector, respectively. The wall tangential vector **t** is used to calculate the wall shear stress τ_w. How it is done will be explained later. At the inflow the velocity was set to $1\frac{m}{s}$ in the streamwise direction, the pressure to zero gradient and $\hat{\nu} = 6.93 \times 10^{-5}\frac{m^2}{s}$. At the outlet



(a) $\frac{U}{U_b}$



(b) $\frac{\nu_t}{\nu}$



(c) f_{r1}

Figure 3. Velocity scaled by the bulk velocity $\frac{U}{U_m}$ (top left) for the Spalart-Allmaras model (SA) and the Spalart-Allmaras model with rotation and curvature correction (SARC), turbulent viscosity scaled by the molecular viscosity $\frac{\nu_t}{\nu}$ (top right) for the Spalart-Allmaras model (SA) and the Spalart-Allmaras model with rotation and curvature correction (SARC), constant f_{r1} for a Rossby number of 0 (SARC Omega 0) and for a Rossby number of 0.5 (SARC). The results of Shur et al. [1] using the streamline and curvature correction are labeled as Shur 2000 SARC

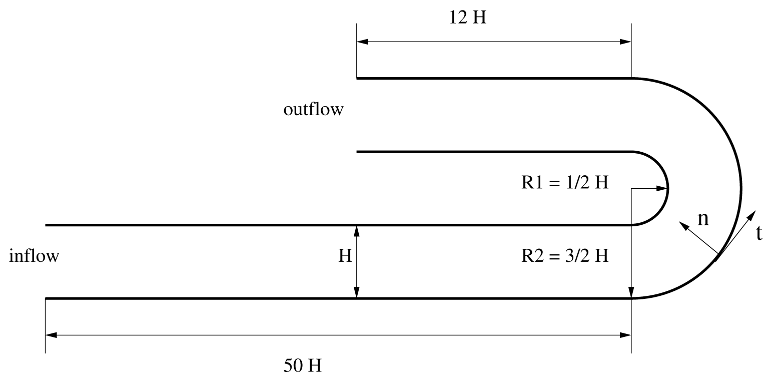
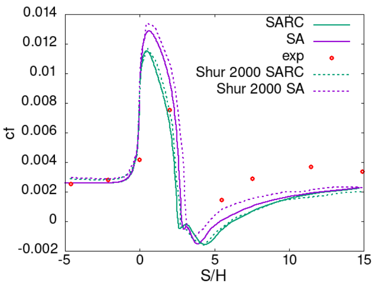


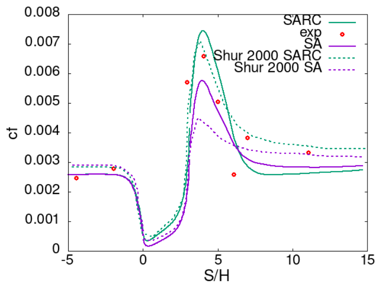
Figure 4. Sketch of the 180° pipe bend configuration

the pressure is set to 0 and the other two variables to zero gradient. At the wall a nutLowReWallFunction was used for ν_t, a no-slip condition for the velocity and a zero gradient condition for the pressure. The Reynolds number based on the mean velocity and the channel height H was $Re = \frac{UH}{\nu} = 10^6$. The mesh had 350 cells in the wall-normal direction and 290 in the streamwise direction. It was ensured that the maximum size of the cells adjacent to the wall was small enough to ensure that the condition $y^+ < 1$ was satisfied. The incompressible solver simpleFoam was used to simulate this test case.

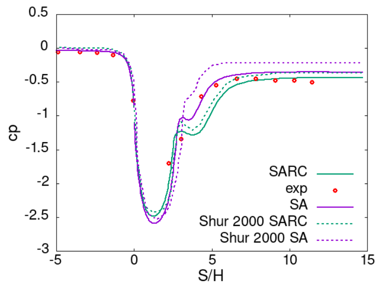
Figure 5 show the computed friction $c_f = \frac{\tau_w}{0.5\rho U_m^2}$ (top row) and pressure $c_p = \frac{p - p_{-5S/H}}{0.5\rho U_m^2}$ (bottom) coefficients. The wall shear stress τ_w is calculate by calculating the dot product of the wall shear stress vector $\boldsymbol{\tau_w}$ output by OpenFOAM at the wall boundaries with the wall tangential vector $\mathbf{t}$ (see Fig. 4), i.e. $\tau_w = -\boldsymbol{\tau_w} \cdot \mathbf{t}$. $p_{-5S/H}$ is the pressure at a distance $S/H = -5$ from the start of the pipe bend. U_m is the bulk velocity. On the right column the results on the inner wall are shown and on the left column the results on the outer wall are visualized. The results of Shur et al. [1] using the Spalart-Allmaras (Shur



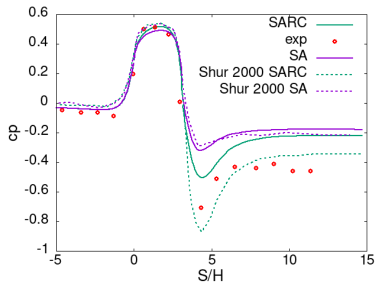
(a) c_f inner wall



(b) c_f outer wall



(c) c_p inner wall



(d) c_p outer wall

Figure 5. Comparison of the friction coefficient c_f (top row) and pressure coefficient c_p (bottom row) for the Spalart-Allmaras model (SA) and the Spalart-Allmaras model with rotation and curvature correction (SARC). The red dots represent the reference experiments. The left column are the results obtained at the inner wall and the right column are the results obtained at the outer wall.

2000 SA) and also the Spalart-Allmaras model with rotation rate and curvature correction (Shur 2000 SARC) are also included. The results obtained by Shur et al. [1] are displayed as dashed lines with the same color code used for the present results. It is evident that the SARC model agrees slightly better with the reference experiments. Inside the outer radius of the pipe bend the friction coefficient c_f is higher for the SARC model compared to the SA model. This is consistent with the high factor f_{r1} observed at the concave side of the bend (see Fig. 6). A high factor f_{r1} increases the turbulence production with respect to the standard SA model and therefore should lead to higher friction at the outer part of the pipe bend. The agreement with the reference simulation of Shur et al. [1] is reasonably good for both models compared, i.e. the standard Spalart-Allmaras model SA and the Spalart-Allmaras model with rotation rate and curvature correction SARC. The reason why the current results do not match exactly those computed by Shur et al. [1] is not known.

Figure 6 shows the computed values for the factor f_{r1} in the pipe bend. It is evident that in the inner (convex) part of the bend the factor is negative and in the outer (concave) part it is positive. This means that close to the convex part of the pipe bend the turbulence will be damped, whereas it will be augmented at the concave part. Similar results are also found in the DNS of Moser and Moin [2] which simulated a weakly curved periodic channel. This means that the current implementation of the model should qualitatively reproduce the influence of curvature on turbulence.

3.4. **Transonic delta wing.** Figure 7 shows a sketch of the setup of the simulation around a delta wing. The second international vortex flow experiment (VFE-2) wing consists of a swept wing with a root chord c of 0.4905 m. The lines delimiting the box are much closer to the wing in the sketch compared to the real setup. The details of the experimental setup together with the measurement results can be found in Chu and Luckring [9]. The domain consists of a rectangular box where the edges are located $18c$ from the wing nose. This domain size should reduce the influence of the artificial boundary conditions on the results to a minimum. On the side of the box which is located at the wing root, a symmetry boundary condition is applied. This allows to half the grid size. For all other sides an inflow-outflow boundary conditions is used for all quantities expect the pressure. For the pressure, an outflow-inflow boundary condition is used. Here the zero gradient boundary condition is used for incoming flow and a fixed value for outgoing flow. The free stream mach number is set to Ma = 0.84 and

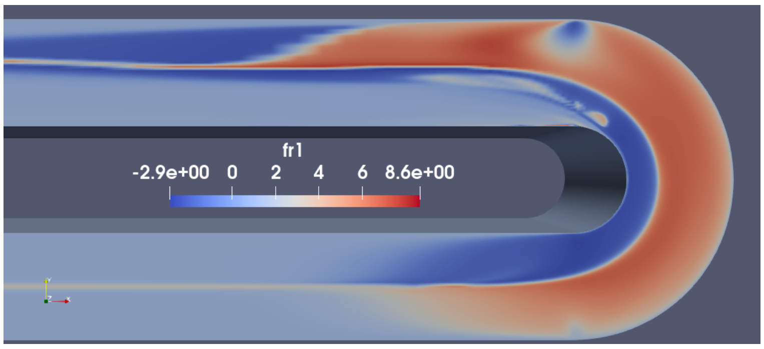


Figure 6. Contour plot of the factor f_{r1}.

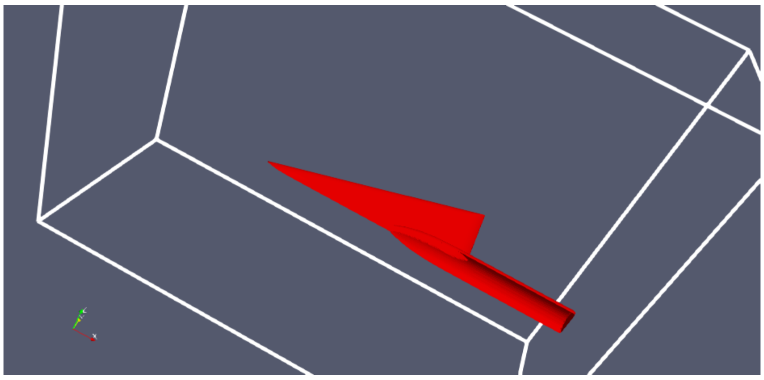


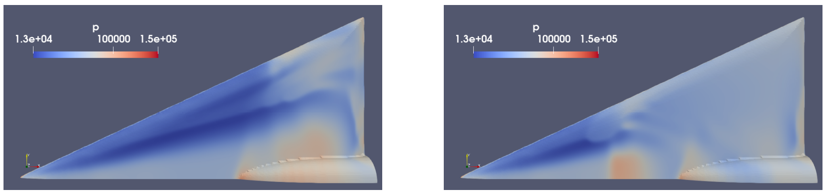
Figure 7. Sketch of the setup used to simulate the flow around the delta wing

the angle of attack is set to $\alpha = 26.7°$. The Reynolds number based on the chord length was approximately Re $= 5 \times 10^6$. For the turbulent viscosity ν_t the wall function nutUSpaldingWallFunction was used in order to avoid a very fine mesh close to the surface. The available wing surface data in tecplot format can be downloaded here: https://www.kbwiki.ercoftac.org/w/index.php/Description_AC1-09#Flow_Domain_Geometry. The file can be converted with paraview to an stl file which is required by snappyHexMesh. The pressure based compressbile solver rhoPimpleFoam with a local time stepping was used to simulate this case.

Five refinement regions are chosen to gradually increase the resolution from the coarsest level to the finest level around the wing. The initial grid had hexagonal cells of 2.4 times the chord length. Note that it is very important to start with cells which are cubes to get a high wall layer coverage near the surface if wall layers are used. The fifth refinement level has a cell side length of 0.07 times the chord length. The surface around the wing is additionally refined to capture the fine curvature at the wing nose and the tail. For the surface a refinement level 8 and 9 was used. A refinement level of 8 correspond to 0.02 times the chord and a refinement level of 9 correspond to a cell size of 0.01 times the chord. In order to resolve the tip vortices well, 9 cells between levels were chosen in order to have a fine resolution close to the wing. This strategy results in a mesh of approximately 7.5×10^5 cells. Five layers are used to increase the wall-normal resolution to capture the sharp velocity gradients near the wall. The layer coverage is roughly 97%.

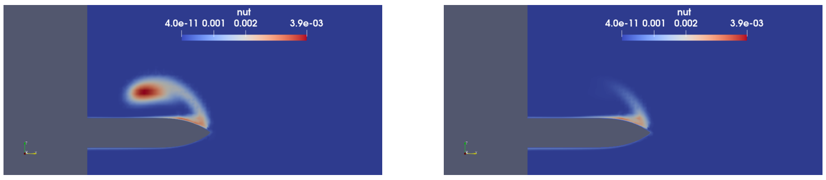
Figure 8 shows the pressure distribution on the upper surface of the delta wing when using the SA model (left) and the SARC model (right). It is evident that when using the SA model the wing tip vortices extend over almost the whole wing chord length. The wing tip vortices lead to a region of low pressure close to the outer part of the wing. These low pressure regions are visible in the pressure contour plot. When using the SARC model the vortices break down at around a third of the chord. When using the SARC the low pressure region caused by the wing tip vortices is suddenly interrupted by a strong pressure increase close to the symmetry plane. The correct prediction of the location of the vortex breakdown is critical for the correct prediction of the lift generated by the wing. For a delta wing configuration the low pressure regions induced by the wing tip vortices are responsible for the major part of the lift generation.

Figure 9 shows a contour plot of the turbulent viscosity ν_t distribution at a plane at $\frac{x}{c} = 0.3$. The left figure shows the prediction using the SA model and the right the prediction of the SARC model. It is evident that when applying the SA model, ν_t obtains high values in the vortex core. This is not the



(a) pressure SA **(b)** pressure SARC

Figure 8. Comparison of the pressure distribution at the surface of the Delta Wing. Left: Spalart-Allmaras model (SA), right: Spallart-Allmaras with Rotation correction (SARC)

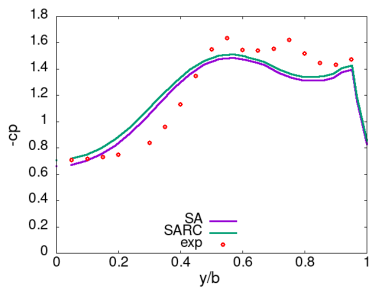


(a) ν_t SA **(b)** ν_t SARC

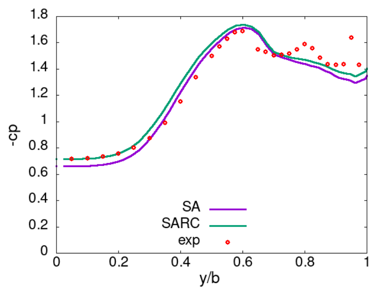
Figure 9. Comparison of the turbulent viscosity ν_t distribution at a plane at $\frac{x}{c} = 0.3$. Left: Spalart-Allmaras model (SA), right: Spallart-Allmaras with Rotation correction (SARC)

case for the SARC which leads to considerable lower values of the turbulent viscosity in the vortex core. Note that the same observation was done by Duraisamy and Iaccarino [13] when studying the influence of the curvature correction applied to the $v^2 - f$ model: When analyzing the turbulent viscosity ν_t in the core of a wing tip vortex, Duraisamy and Iaccarino [13] found that the $v^2 - f$ model without curvature correction predicted unrealistic high values of ν_t. These high values of ν_t were in contradiction with the stabilizing effect of a vortex exhibiting a nearly solid body rotation. After including a curvature correction to the $v^2 - f$ model, the value of ν_t in the vortex core observed by Duraisamy and Iaccarino [13] dropped considerably reaching realistic values.

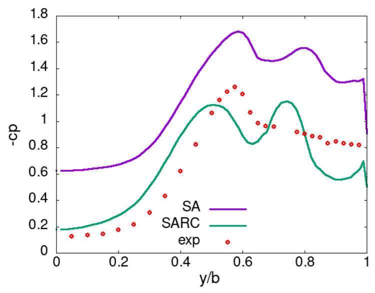
Figure 10 shows the $-c_p$ values over the upper part of the wing span (i.e. $\frac{z}{b} \geq 0$) at different position $\frac{x}{c}$ along the wing chord. The bottom right figure in Fig. 10 shows the $-c_p$ values at the at symmetry plane (i.e. $\frac{y}{b} = 0$). The constant b represents the local wing span. The constant c is the chord length at the symmetry plane. The bottom right figure shows the distribution of $-c_p$ along the upper part of the wing along the chord at the symmetry plane. The data used for the plotting are extracted from paraview. I proceeded like the following to extract the data: First only the pressure at the wing surface is retained in paraview. After that we have to slice the wing surface at the desired position. As last point one has to save the date with a csv format (thanks to the reviewer suggesting how to extract the data from paraview for the purpose making a line plot). A python file which does the extraction of the data automatically is added to the case files. It is evident that at the first measurement position $\frac{x}{c} = 0.2$ both models give the same pressure distribution along the wing span. The peak of $-c_p$ is not as pronounced as visibly in the experiment. The reason is the low resolution used in the simulation. When doubling the points in all three directions in the provided blockMeshDict, the pressure peak at $\frac{x}{c} = 0.2$ is higher and agrees well with the reference experiments. The results are not shown for sake of brevity. In this work we want to highlight the differences in the flow field obtained when using the SA and the SARC model and not making a mesh sensitivity study. The interested readers are free to modify the provided cases as they wishes. The results of both models start to deviate from a position $\frac{x}{c} = 0.6$ along the chord. For the SARC model the strength of the wing tip vortices starts to diminish. This is visible by the lower $-c_p$ value close to the wing tip. For the SA model the $-c_p$ value is higher. Higher $-c_p$ values mean lower pressure close to the vortex core. Low pressure in the vortex core is an indicator for large turning speeds. At a position of $\frac{x}{c} = 0.8$ the SARC already predicts the absence of the wing tip vortex while it is still present for the SA model and also in the reference experiments. The agreement with the reference experiment is much better for the SARC compared with the SA model. At a position of $\frac{x}{c} = 0.95$ the agreement of the SARC with the experiment is also much better compared to the SA model. Interesting is the comparison of the evolution of $-c_p$ along the upper surface adjacent to the symmetry line. The $-c_p$



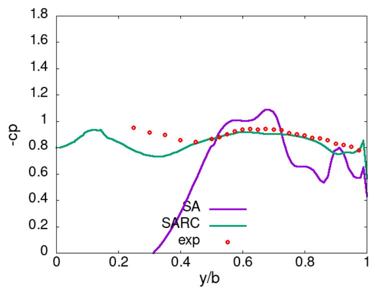
(a) $-c_p$ at $\frac{x}{c} = 0.2$



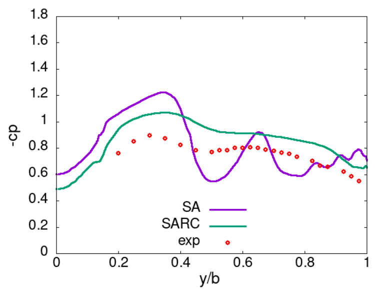
(b) $-c_p$ at $\frac{x}{c} = 0.4$



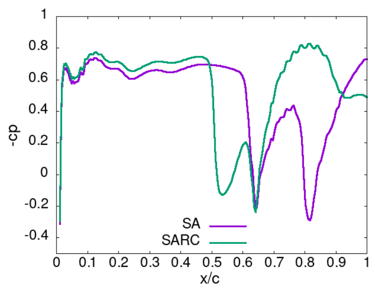
(c) $-c_p$ at $\frac{x}{c} = 0.6$



(d) $-c_p$ at $\frac{x}{c} = 0.8$



(e) $-c_p$ at $\frac{x}{c} = 0.95$



(f) $-c_p$ at $\frac{y}{c} = 0.0$

Figure 10. Comparison of the pressure coefficient $-c_p$ at different positions along the cord and at the bottom right the values of $-c_p$ at a line in streamwise direction at the symmetry plane is sown.

for the SARC experiences a first sharp drop at a position of $\frac{y}{c} = 0.4$. This means that a first occurrence of a shock is predicted by the simulation using the SARC turbulence model. This shock wave is not predicted if we use the SA model. The second shock occurs at the intersection of the wing and the shaft holding the wing located at around $\frac{y}{c} = 0.6$.

4. Conclusion

In this technical note, an implementation of the rotation and streamline curvature correction to the Spalart-Allmaras model (SARC) as suggested in Shur et al. [1] is presented. The motivation for the present work is that no publicly available repository was found which contains the full set of terms suggested in Shur et al. [1]. The equations implemented are also found in the NASA turbulence model resource homepage. The correctness of the implementation is assessed by means of three different test cases: The 2D flow of a bump in a channel, a periodically rotating channel and the flow in a U-turn. The model mimics correctly the influence of rotation and stream line curvature on turbulence.

Regarding the 2D bump flow, the results of the current implementation of the model of Shur et al. [1] agree very well with the results obtain with the implementation in the code CFL3D.

Regarding the periodically rotating channel flow and the flow in a U-turn we see quantitative differences between the results of the current implementation and the results published in the paper of Shur et al. [1]. For the rotating channel flow the turbulence viscosity ν_t predicted with the current implementation is higher compared to the results of Shur et al. [1]. Also for the distribution of the term f_{r1} over the channel height we found differences between the current implementation and the results of Shur et al. [1].

Regarding the flow in a U-turn we also observe quantitative differences in the evolution of the skin friction and pressure coefficient at the walls. Unfortunately we could not spot the reason of this discrepancy with clarity. A consistency check regarding the term f_{r1} for the rotating channel flow provided in the appendix showed that the current implementation of the equation provided by Shur et al. [1] is clearly different compared to the implementation of the authors of the paper. The current implementation of f_{r1} is however consistent with the analytical equation derived for f_{r1} in case of the rotating channel flow. For this reason, we can state with a high degree of confidence, that the current implementation of the equations provided by Shur et al. [1] is correct. Since the source code along with all test cases is publicly available, I encourage every interested reader to help to definitely solve these issues.

Finally a 3D transonic flow around a delta wing is tackled. It could be shown that the shock induced vortex breakdown happens further upstream if the Spalart-Allmaras model with curvature correction (SARC) is applied. In comparison the vortex break down predicted when using the standard Spalart-Allmaras (SA) model as implemented in the official OpenFOAM release, happens further downstream. The SARC model leads to much smaller turbulent viscosity ν_t in the core of the wing tip vortex compared to the SA model. Small ν_t values are more consistent with the turbulence damping effect of solid-body-like vortices than high values of ν_t as computed by the SA model. The high values of ν_t in the SA are generated by high shear values.

Acknowledgements

I would like to thank all reviewers for all their comments and suggestions. With their help the quality of the article improved significantly compared to the first submitted version. I actually see the process of publishing an article using an open source code like OpenFOAM together with case files accessible to the reviewers (when published of course also to the general public) as a joint effort between the authors of the article and the reviewers with the aim to deliver a best possible contribution to the community. Many thanks again to the reviewers for their effort. I would like to thank also the reviewer pointing to the bug present in the initial submission and providing a suggestion how to correct the bug. The reviewer provided a suggestion (which is actually implemented in the provided case files) how to correctly retrieved the index of the moving reference zone.

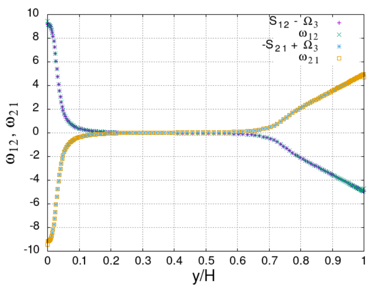
Author Contributions: Conceptualisation, M.A.; methodology, M.A.; software, M.A; validation, M.A.; formal analysis, M.A.; investigation, M.A.; resources, M.A.; data curation, M.A.; writing—original draft preparation, M.A.; writing—review and editing, M.A.; visualisation, M.A.; supervision, M.A.; All authors have read and agreed to the published version of the manuscript.

Appendix A. Consistency check of the implemented equations

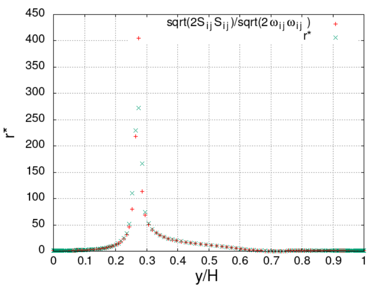
In this section we will perform a consistency check of the implemented equations using the rotating channel case solutions. The rotating channel is a 1D flow and the resulting tensors and the expressions derived from it are relatively simple. For this reason, it is an ideal case to test the consistency of the implemented equations. Note that the gnuplot and python files used to produce all plots used for the consistency check are available in the provided case files under the directory of the rotating channel test case. For this reason the interested reader can verify the correctness of the implemented equation. We are starting our analysis with the definition of the shear stress tensor S_{ij} and the rotation rate tensor ω_{ij} in the 1D rotating channel case (see also Eqn. (6)):

$$S_{ij} = \begin{bmatrix} 0 & S_{12} & 0 \\ S_{21} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; \omega_{ij} = \begin{bmatrix} 0 & S_{12} - \Omega_3 & 0 \\ -S_{21} + \Omega_3 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; S_{12} = S_{21} = \frac{1}{2} u_y \tag{10}$$

u_y is the derivative of the streamwise component of the velocity u with respect to the y-coordinate. We see that the rotation rate tensor ω_{ij} is a function of the velocity gradient $\frac{1}{2}u_y$ and the rotation rate Ω_3 around the z-axis. The first step of the consistency check is to write the tensors S_{ij} and ω_{ij} to the hard drive. After that we check if the components of the tensor ω_{ij} can be retrieved as function of $\frac{1}{2}u_y$ and Ω_3. Figure 11 on the left visualizes this consistency check. We see that the components of the tensor ω_{ij} can be retrieved as function of S_{12}, S_{21} and Ω_3. Looking at Eqn. (10) we see that the rotation rate tensor ω_{ij} should become zero in the inner part of the channel where the slope of the velocity $u_y = 2\Omega_3$ (see figure 3). When looking at the left figure of figure 11 wee that this is actually the case underlining the correctness of the computation of ω_{ij}. The figure on the right of figure 11 displays the consistency check of r^* vs. ω_{ij} and S_{ij}. See also Eqn. 5 for the definition of r^*. For the rotating channel flow r^* can be written as

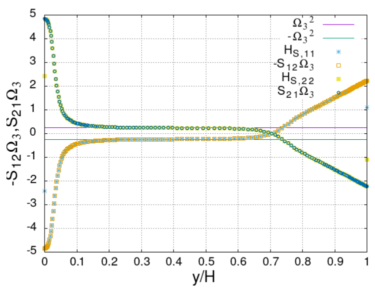


(a) check ω_{ij} vs. S_{ij}

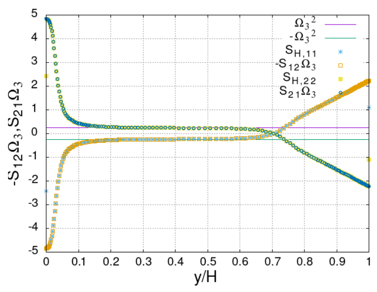


(b) check r^* vs. ω_{ij}, S_{ij}

Figure 11. Consistency check of ω_{ij} vs. S_{ij} (left) and r^* vs. ω_{ij}, S_{ij}



(a) check $\epsilon_{imn}S_{jn}\Omega_m$ vs. S_{ij} and Ω_m



(b) check $\epsilon_{jmn}S_{in}\Omega_m$ vs. S_{ij} and Ω_m

Figure 12. Consistency check of $\epsilon_{imn}S_{jn}\Omega_m$ (left) and $\epsilon_{jmn}S_{in}\Omega_m$ (right) vs. S_{ij} and Ω_m

$$r^* = \frac{S}{\omega} = \frac{\sqrt{2(S_{12}S_{12} + S_{21}S_{21})}}{\sqrt{2(\omega_{12}\omega_{12} + \omega_{21}\omega_{21})}}. \tag{11}$$

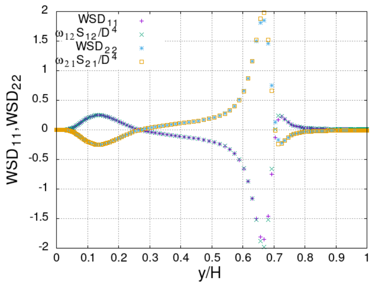
We see from Eqn. (11) that the volScalarField r^* can be retrieved from the components of the volTensorFields ω_{ij} and S_{ij} written to the hard drive. From Fig. 11 on the right we see that r^* can be retrieved from the components of ω_{ij} and S_{ij}. What we can observe is also that r^* is always positive and it obtains very high values in the middle of the channel. The observations that r^* is always positive is consistent with the definition of r^*: it is a ratio of two norms. When looking at the equation for ω_{ij} of the channel case (see Eqn. (10)) it is evident that the components of ω_{ij} are becoming zero for a velocity gradient $u_y = 2\Omega_3$. For both components of ω_{ij} equal to zero the denominator of r^* becomes also zero and hence r^* becomes very large. The behaviors of always positive r^* values and the large values in the region of $u_y = 2\Omega_3$ are reflected in the Fig. 11 on the right.

The next step is to write down the tensors $\epsilon_{imn}S_{jn}\Omega_m$ and $\epsilon_{jmn}S_{in}\Omega_m$ for the case of the rotating channel flow. We can easily verify that following relations holds for the two latter tensors in case of the 1D rotating channel flow:

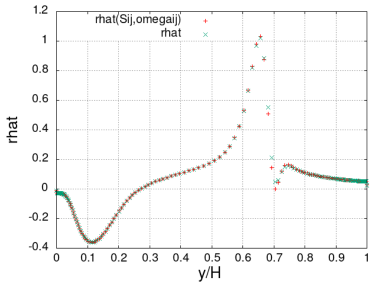
$$\epsilon_{jmn}S_{in}\Omega_m = S_{H,ij} = \epsilon_{imn}S_{jn}\Omega_m = H_{S,ij} = \begin{bmatrix} -S_{12}\Omega_3 & 0 & 0 \\ 0 & S_{21}\Omega_3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{12}$$

Figure 12 shows the consistency check of $\epsilon_{imn}S_{jn}\Omega_m = H_{S,ij}$ on the left and $\epsilon_{jmn}S_{in}\Omega_m = S_{H,ij}$ vs. S_{ij} and Ω_m on the right. It is evident that the components of the tensors $\epsilon_{imn}S_{jn}\Omega_m$ and $\epsilon_{jmn}S_{in}\Omega_m$ can be retrieved as function of S_{ij} and Ω_3. The lines with constant values of Ω_3^2 and $-\Omega_3^2$ are also shown in the plots. It is easy to verify that for the region where the gradient of the velocity is approximately $u_y = 2\Omega_3$ the components of the tensors become equal to $S_{H,12} = S_{H,12} = -\Omega_3^2$ and equal to $S_{H,21} = S_{H,21} = \Omega_3^2$.

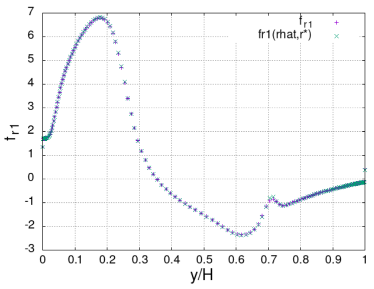
The next step is to write down the tensor $\frac{2\omega_{ik}S_{jk}}{D^4}$ for the case of the rotating channel flow:



(a) check $\frac{2\omega_{ik}S_{jk}}{D^4}$ vs. S_{ij} and ω_{ij}



(b) check $\hat{r}$ vs. $\frac{2\omega_{ik}S_{jk}}{D^4}$



(c) check f_{ri} vs. r^* and $\hat{r}$

Figure 13. Consistency check of $\frac{2\omega_{ik}S_{jk}}{D^4}$ vs. S_{ij} and ω_{ij} (top left), $\hat{r}$ vs. $\frac{2\omega_{ik}S_{jk}}{D^4}$ (top right) and f_{ri} vs r^* and $\hat{r}$ (bottom center).

$$\frac{2\omega_{ik}S_{jk}}{D^4} = WSD_{ij} = \frac{2}{\{0.5[2(S_{12}^2+S_{21}^2)+2(\omega_{12}^2+\omega_{21}^2)]\}^2}\begin{bmatrix}\omega_{12}S_{12} & 0 & 0\\ 0 & \omega_{21}S_{21} & 0\\ 0 & 0 & 0\end{bmatrix} \tag{13}$$

For better readability we define the tensor $\frac{2\omega_{ik}S_{jk}}{D^4} = WSD_{ij}$. Figure 13 on the top left shows the consistency check $\frac{2\omega_{ik}S_{jk}}{D^4}$ vs. S_{ij} and ω_{ij}. It is evident the the volTensorField $\frac{2\omega_{ik}S_{jk}}{D^4} = WSD_{ij}$ written to the hard drive can be retrieved from the two volTensorFields S_{ij} and ω_{ij}. The next tensors used for the consistency check are defined as follows: $\epsilon_{imn}S_{jn}\Omega_m = H_{S,ij}$ and $\epsilon_{jmn}S_{in}\Omega_m = S_{H,ij}$. For the rotating channel flow now we can compute $\hat{r}$ as function of WSD_{ij}, $H_{S,ij}$ and $S_{H,ij}$ (note that the tensor $\frac{DS_{ij}}{Dt}$ in Eqn. (9) is zero for the rotating channel flow):

$$\hat{r} = WSD_{ij}(H_{S,ij}+S_{H,ij}) = WSD_{12}(H_{S,12}+S_{H,12}) + WSD_{21}(H_{S,21}+S_{H,21}) \tag{14}$$

Figure 13 on the top right compares the volScalarField $\hat{r}$ written to the hard drive with the quantity using the volTensorFields WSD_{ij}, $H_{S,ij}$ and $S_{H,ij}$ computed by means of Eqn. (14). It is evident that they are identical. The last check shown in Fig. 13 on the bottom center is the consistency check f_{ri} vs. r^* and $\hat{r}$. It means that the volTensorField f_{r1} is plotted against the relation obtained using the volScalarFields r^* and $\hat{r}$ inserted into Eqn. (4). Also in this case the two quantities are identical.

Summing up, we did a consistency check of all terms involved in the computation of f_{r1} and did not find any inconsistencies. Hence, we can deduce with a high degree of confidence, that the equations are implemented correctly. Since we did not find any inconsistencies, we can deduce that the current implementation of the equations shown in Shur et al. [1] is different from the one of the authors of the article referenced.

In order to confirm the above mentioned hypothesis, we will check in the following if our implementation of the equations provided by Shur at al. [1] is the same as the implementation done by the authors of the article. As already mentioned above, the term f_{r1} is a function of the velocity gradient u_y and the rotation rate Ω_3. In order to compute f_{r1} as function of the velocity, the reader can extract the velocity profile computed by Shur et al. [1] from the article or use the data provided in the case files. The data provided with the case are extracted manually from the paper with a web program and are a bit wiggly. After that, the interested reader can use the velocity extracted from the paper to check if the term f_{r1}

as plotted by Shur et al. [1] can be retrieved from the velocity shown in the same paper. In the case files there is a python script which performs this task and implements the analytical equations Eqns. (10)-(14). The results of this computations are shown in Fig. 14. In the figure we see f_{r1} computed using the velocity of the current implementation together with the python script (red curve), f_{r1} computed by the turbulence model inside OpenFOAM (green curve), f_{r1} computed using the velocity extracted from the paper by Shur et al. [1] together with the python script (blue curve) and f_{r1} extract directly from the paper by Shur et al. [1] (black curve). We see that the red and green curve coincide very well. This means that the implementation within OpenFOAM is consistent with the analytical relation derived for a rotating channel case (see Eqns. (10)-(14)). We can also conclude that the current implementation differs from the implementation of Shur et al. [1] since the blue and black curve do not coincide. The exact reason why the present f_{r1} term is different from the one computed by Shur et al. [1] can not be definitely clarified. The interested reader may help to solve this issue.

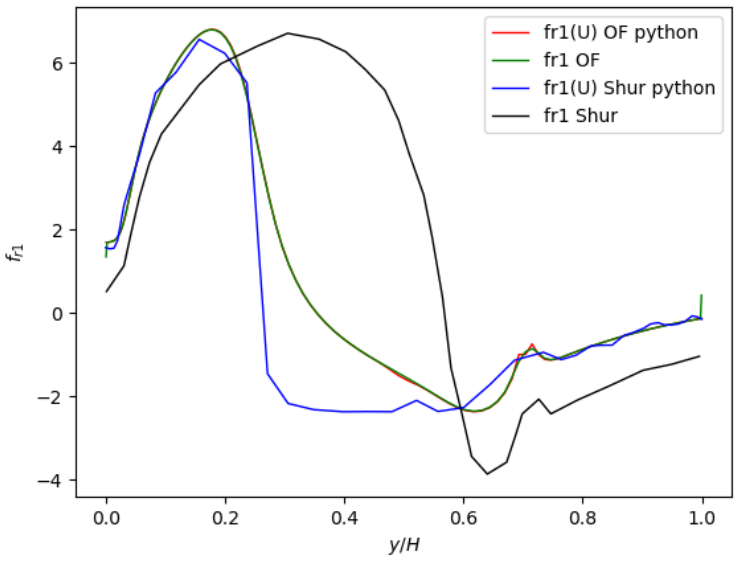


Figure 14. Comparison of the term f_{r1} computed using the velocity of the current implementation together with a python script (red curve), f_{r1} computed by the turbulence model inside OpenFOAM (green curve), f_{r1} computed using the velocity extracted from the paper by Shur et al. [1] together with the same python script as for the red curve (blue curve) and f_{r1} extract directly from the paper by Shur et al. [1] (black curve)

References

[1] M. L. Shur, M. K. Strelets, A. K. Travin, and P. R. Spalart, “Turbulence modeling in rotating and curved channels: assessing the spalart-shur correction,” *AIAA journal*, vol. 38, no. 5, pp. 784–792, 2000.

[2] R. D. Moser and P. Moin, “The effects of curvature in wall-bounded turbulent flows,” *Journal of Fluid Mechanics*, vol. 175, pp. 479–510, 1987.

[3] G. Brethouwer, P. Schlatter, and A. V. Johansson, “Effects of rapid spanwise rotation on turbulent channel flow with a passive scalar,” in *Seventh International Symposium on Turbulence and Shear Flow Phenomena*. Begel House Inc., 2011.

[4] G. Brethouwer, “The effect of rotation on rapidly sheared homogeneous turbulence and passive scalar transport. linear theory and direct numerical simulation,” *Journal of Fluid Mechanics*, vol. 542, pp. 305–342, 2005.

[5] P. Spalart and M. Shur, “On the sensitization of turbulence models to rotation and curvature,” *Aerospace Science and Technology*, vol. 1, no. 5, pp. 297–302, 1997.

[6] Y. H. Alahmadi and A. F. Nowakowski, “Modified shear stress transport model with curvature correction for the prediction of swirling flow in a cyclone separator,” *Chemical Engineering Science*, vol. 147, pp. 150–165, 2016.

[7] Y. Ren, Z. Zhu, D. Wu, J. Mu, and X. Li, “An improved turbulence model for separation flow in a centrifugal pump,” *Advances in Mechanical Engineering*, vol. 8, no. 6, p. 1687814016653310, 2016.

[8] Y. You, F. Seibold, S. Wang, B. Weigand, and U. Gross, “Urans of turbulent flow and heat transfer in divergent swirl tubes using the k-ω sst turbulence model with curvature correction,” *International Journal of Heat and Mass Transfer*, vol. 159, p. 120088, 2020.

[9] J. Chu and J. M. Luckring, “Experimental surface pressure data obtained on 65 deg delta wing across reynolds number and mach number ranges. vol. 1: Sharp leading edge,” Tech. Rep., 1996.

[10] B. Chaouat, “Simulations of turbulent rotating flows using a subfilter scale stress model derived from the partially integrated transport modeling method,” *Physics of Fluids*, vol. 24, no. 4, p. 045108, 2012.

[11] G. Brethouwer, “Statistics and structure of spanwise rotating turbulent channel flow at moderate reynolds numbers,” *Journal of Fluid Mechanics*, vol. 828, pp. 424–458, 2017.

[12] D. Monson, H. Seegmiller, and P. McConnaughey, “Comparison of experiment with calculations using curvature-correctedzero and two equation turbulence models for a two-dimensional u-duct,” in *21st Fluid Dynamics, Plasma Dynamics and Lasers Conference*, 1990, p. 1484.

[13] K. Duraisamy and G. Iaccarino, “Curvature correction and application of the v2-f turbulence model to tip vortex flows,” *Center for Turbulence Research Annual Research Briefs*, pp. 157–168, 2005.