Overview of the Digital Filter Implementation in *FDL3DICE*

David R. González*

Naval Surface Warfare Center, Indian Head, MD 20640-5035

Arvind T. Mohan† and Datta V. Gaitonde‡

The Ohio State University, Columbus, OH, 43210

This report documents the implementation of digital filtering into *FDL3DICE*. The approach taken is based on Touber & Sandham's two-dimensional technique. However, it has been extended to function in computational space in order to accomodate the curvilinear meshes required for more general CFD analyses. Digital filtering was initially conceived to work in an equi-spaced Cartesian grid. A common approach taken in the literature to address non-equispaced grids is to develop the inflow turbulence in a Cartesian grid and subsequently interpolate the filtered data onto the analysis mesh. By working in computational space, the exact digital filter formulation of Touber & Sandham can be applied and the results can subsequently be transformed to physical space and imposed on the inflow mesh without requiring interpolation. The technique used in the current implementation has been tested on a supersonic-jet model simply to verify its proper functionality. Complete simulations to test the filter have yet to be conducted.

I. Introduction

Turbulent inflow conditions for direct and large-eddy simulations continue to be a critical subject of research in the fluid dynamics community. Depending on the problem being investigated, high-quality turbulent inflow conditions that accurately capture the dynamics of the flow under investigation can significantly reduce the computational efforts required for a successful simulation. Common approaches have included recycling/rescaling and synthetic turbulence, among others. This report is concerned with the latter. Here, we document the implementation of a digital filtering (DF) routine into *FDL3DICE*. While the technique has been implemented for a supersonic jet case, the routine has been generalized so that minimal user modifications are required to apply the technique to other cases.

The following sections present a brief overview of the digital filtering technique as well as some preliminary results showing the implementation of the DF. At this time, full simulations to validate the approach have yet to be conducted. This report is aimed at discussing the specifics of its use within *FDL3DICE* for future reference.

II. Overview of Digital Filtering

The digital filtering approach implemented in *FDL3DI* follows closely the work of Touber & Sandham.[1,2] Consider the sequence of p random numbers with *zero-mean* and *unit variance* given in Eq. 1.

$$\{r_k\}_{1\leq k\leq p}. \tag{1}$$

A discrete operator, F_N, can be defined as:

[*] Mechanical Engineer, Warhead & Propulsion Technology Branch, 4103 Fowler Rd., Bldg. 302 Ste. 107, Senior Member AIAA.

[†] PhD Student, Mechanical and Aerospace Engineering, Student Member AIAA.

[‡] John Glenn Chair Professor, Mechanical and Aerospace Engineering, Fellow AIAA.

$$v_k \equiv F_N(r_k) = \sum_{j=-N}^{N} b_j r_{k+j} \tag{2}$$

where N is a positive integer and $\{b_j\}_{-N \leq j \leq N}$ are filter coefficients to be defined. As described in Touber & Sandham,[1] the filtering operation in Eq. 2 is linear and it can be shown that the correlation between the filtering coefficients is given by:

$$\overline{v_k v_{k+q}} = \sum_{j=-N+q}^{N} b_j b_{j-q}. \tag{3}$$

To impose the desired *spatial* correlation on the random number sequence, the two-point correlation function is modeled as:

$$R(x_k + x) = exp\left(-\frac{\pi x}{2 I_x}\right) \tag{4}$$

where x_k is a reference location, x is some point away from the reference, and the integral length scale is I_x. Note that Eq. 4 is one-dimensional for simplicity. Extension to multiple dimensions follows simply by developing a separate series of filter coefficients as in Eq. 2 and correlating in these in space via Eq. 4.

A. Specification of Integral Length Scales

An important aspect of the DF is the need to specify the integral lengths scales associated with the turbulent flow under study. In the seminal papers of Touber & Sandham,[1,2] Xie & Castro,[3] and Klein *et.al.*,[4] the integral length was defined on a computational in terms of a *constant* spacing, Δx, and an integer, n, number of grid points. With such a specification, the integral length scale became $I_x = n\Delta x$ and the disctance from the reference location $x = q\Delta x$. With these definitions, the one-dimensional two-point correlations were specified as:

$$R(x_k + q\Delta x) \equiv \frac{\overline{v_k v_{k+q}}}{\overline{v_k v_k}} = exp\left(-\frac{\pi q}{2n}\right). \tag{5}$$

B. Calculation of the Convolution Coefficients, b_k

Substituting Eq. 3 into the right-hand-side of Eq. 5, the unknown filter coefficients, b_k, can be computed by solving the system:

$$\frac{\sum_{j=-N+q}^{N} b_j b_{j-q}}{\sum_{j=-N}^{N} b_j^2} = exp\left(-\frac{\pi q}{2n}\right). \tag{6}$$

The solution to Eq. 6 can be approximated as:

$$b_k \approx \frac{\widetilde{b}_k}{\left(\sum_{j=-N}^{N} \widetilde{b}_j^2\right)^{1/2}} \Rightarrow \widetilde{b}_k = exp\left(-\frac{\pi k}{n}\right) \tag{7}$$

as long as $N \geq 2n$, as pointed out by Klein, *et.al.*[4]

With the operations given in Eqs. 2-7, one obtains a field of numbers containing the desired coherence from the initial zero-mean, unit-variance random numbers generated in Eq. 1. As mentioned previously, the one-dimensional convolution coefficients, b_k, can be extended to multiple dimensions by simply defining a two-dimensional convolution of the form $b_{jk} = b_j b_k$.

C. Temporal Correlation

The main success of Xie & Castro's[3] work was in terms of simplifying Klein, *et.al.*'s[4] three-dimensional filtering into a two-dimensional problem. This was accomplished by adopting the following formula:

$$\rho_k = v_k^{old} exp\left(-\frac{\pi \Delta t}{2\tau}\right) + v_k \sqrt{1 - exp\left(-\frac{\pi \Delta t}{\tau}\right)}, \tag{8}$$

where Δt is the assumed time step, τ is a Lagrangian time scale ($\tau = I_x/U_{loc}$), and I_x and U_{loc} are the prescribed integral length scale and a local streamwise velocity scale, respectively. Once the operation in Eq. 8 is conducted, the random number sequence contains all the *spatial* and *temporal* correlations requested by the user.

D. Reynolds Stress Distribution

The final aspect of the digital filtering is the imposition of the single-point correlations, or Reynolds stresses. These Reynolds stresses must be supplied to the DF routine *a priori* and depend heavily on the flow under investigation. These correlations are then used in the Cholesky decomposition first proposed by Lund, *et.al.*[5] as:

$$\begin{bmatrix} u(0,y,z,t) \\ v(0,y,z,t) \\ w(0,y,z,t) \end{bmatrix} = \begin{bmatrix} \langle u(0,y,z) \rangle \\ \langle v(0,y,z) \rangle \\ \langle w(0,y,z) \rangle \end{bmatrix} + \begin{bmatrix} \sqrt{R_{11}} & 0 & 0 \\ R_{21}/\sqrt{R_{11}} & \sqrt{R_{22} - (R_{21}/\sqrt{R_{11}})^2} & 0 \\ 0 & 0 & \sqrt{R_{33}} \end{bmatrix} \begin{bmatrix} \rho^u(y,z) \\ \rho^v(y,z) \\ \rho^w(y,z) \end{bmatrix}. \tag{9}$$

In Eq. 9, the first term on the right-hand-side corresponds to the *mean* turbulent velocity profile while the second term represents the turbulent fluctuations. The components of the Reynods stress tensor used in the Cholesky decomposition above are represented by $\{R_{ij}\}_{(i,j)\in\{1,2,3\}}$. Note that in Eq. 9, the XZ and YZ components of the Reynolds stress tensor are assumed to be negligible and are not accounted for in the decomposition. For completeness, the user is referred to the original paper by Lund, *et.al.*,[5] which includes the full decomposition with all Reynolds stress components.

E. Extension to Compressible Flows

The approach presented in the previous sections is geared towards incompressible flows in which only the velocity fluctuations are relevant and fluctuations in the thermodynamic variables are negligible. For most of the applications of interest in the *High-Fidelity Computational Multi-Physics Lab* (HFCMP), compressible flows are of great importance and a means of specifying the thermodynamic variables from the simulated instantaneous velocity field must developed.

Morkovin[6] first derived such a method to link velocity to temperature fields, referred to as the *Strong Reynolds Analogy* (SRA). Touber & Sandham[1,2] make use of this approach. The link between velocity, u_i', and temperature, T', fluctuations is given in the SRA by:

$$\frac{T'}{\overline{T}} = -(\gamma - 1)Ma^2 \frac{u'}{U} \Rightarrow Ma^2 = M^2 \frac{U^2}{\overline{T}}, \tag{10}$$

where $\overline{T}$ is the local mean temperature given by the Crocco-Busemann relation as:

$$\overline{T}(y) = T_w + (T_\infty - T_w)\frac{U(y)}{U_\infty} + \frac{U(y)}{2C_p}(U_\infty - U(y)). \tag{11}$$

In Eq. 11, $\overline{T}(y)$ corresponds to the mean temperature profile, T_∞ and T_w are the freestream and wall temperatures, respectively, and U is the specified mean velocity profile. The ratio of specific heats at constant pressure is given by C_p. Equation 11, along with the prescribed inflow velocity profile, allows us to approximate the temperature fluctuations via the SRA (Eq. 10).

Another assumption of the SRA is that of negligible pressure fluctuations. Recalling the theory of boundary layers, it is frequently assumed that the pressure is constant across a boundary layer. Invoking the perfect-gas relations, a link between the temperature and density fluctuations can be assumed as:

$$\frac{\rho'}{\overline{\rho}} = -\frac{T'}{\overline{T}} \tag{12}$$

While Touber & Sandham[1,2] utilized Eq. 10 in their simulations, initial testing showed some development of negative densities, which would cause problems when implemented into *FDL3DICE*. To tackle this issue, research was conducted into potential alternatives to Morkovin's SRA. As a surrogate, Cebeci & Smith[7] developed an *extended* SRA (ESRA) in which they accounted for the heat flux at the wall boundary. Their relationship between the velocity and temperature fluctuations is given by:

$$-\frac{T'/\overline{T}}{(\gamma-1)Ma^2\frac{u'}{U}} = 1 - C_p\frac{T_{t\infty}-T_w}{U_\infty U}, \tag{13}$$

where $T_{t\infty}$ corresponds to the stagnation temperature in the freestream. Equation 13 is the formulation currently implemented in *FDL3DICE*. As with Morkovin's SRA, Eq. 13 has also been shown to suffer from deficiencies. However, for the constant wall temperature case tested while implementing the DF scheme, good results were obtained in that the density and temperature fluctuations remained realizable (*i.e.* greater than zero) throughout the simulations.

It shall be noted that it is possible to modify the ESRA (Eq. 13) for adiabatic conditions at the wall. Future improvements/extensions to the DF routine could focus on implementing such conditions. A relatively recent development is that of a *generalized* Strong Reynolds Analogy developed by Zhang, *et.al.*[8] Such an analogy was found to perform very accurately over numerous wall boundary conditions.

F. Random Number Generator

A cursory, but very important, piece of the DF scheme is that of the pseudo-random number generator. For large-eddy simulations (LES) such as those conducted in the HFCMP lab, solutions must be integrated over a long time period. Therefore, it is imperative to adopt a random number generator with a sufficiently long period in order to avoid imposing any unphysical cyclic behavior into the solution. For this purpose, the Mersenne Twister[9] has been used as it has a period of $2^{19937}-1$, thereby guaranteeing that the periodicity problem is a non-issue.

The Box-Muller theorem is also invoked to guarantee that an independent sequence of random numbers with normal distribution of unit-standard deviaton is obtained. Here, two sequences, a and b, are generated by the Mersenne Twister and combined as $c = \sqrt{-2ln(a)}\cos 2\pi b$ and $d = \sqrt{-2ln(a)}\cos 2\pi b$. These two sequences are then processed through the filtering procedures described in the previous sections.

G. Summary of Digital Filtering Procedure

As a summary, the following describes the steps taken by the solver to generate the turbulent field using the digital filtering approach:

1. Specify filtering inputs
   (a) Set integral length scales, I_x, I_y, I_z in terms of grid points
   - This sets the filter widths as $N = 2n_{I_x}$

   (b) Select a *local* mean velocity, U_{loc}, for the calculation of the Lagrangian time scale
   (c) Specify whether an incompressible or compressible formulation will be used
2. Generate a sequence of random number for each velocity component
   - Decorrelate the sequences with the Box-Muller theorem
3. Generate the convolution coefficients, b_j
4. Filter random number field with the convolution coefficients to impose spatial correlation
5. Process the filtered coefficients with Eq. 8 to impose temporal correlation
6. Use Lund's transform to impose Reynolds-stress distributions on the correlated random number fields

7. If simulating a compressible flow:
   (a) Compute the temperature fluctuations via SRA or derivative
   (b) Compute density fluctuations from perfect-gas relation, assuming a constant pressure throughout the boundary layer
8. Impose fluctuation field on inflow boundary

III. *FDL3DICE* Implementation

A. Extension to Curvilinear Models

As the majority, if not all, of the work that takes place at the HFCMP lab requires the use of grids with variable spacing, an efficient implementation of the DF is required in order to avoid the need for an interpolation at every time step. While the computational effort of the interpolation in a two-dimensional boundary may not be excessive, any savings that could be incorporated into the routine will ultimately benefit the user. For this reason, an approach similar to that of Dhamankar, *et.al.*[10] has been used in the DF implemented in *FDL3DICE*.

Our routine is identical to that described in the previous section with the exception that we assume that all of the filtering takes place in the *computational* domain. We assume that the correlated field of random numbers (Step 5 in §II.G) correspond to *contravariant* velocity components. These are then transformed to physical space prior to imposing Lund's transform by:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} U'x_\xi + V'x_\eta + W'x_\zeta \\ U'y_\xi + V'y_\eta + W'y_\zeta \\ U'z_\xi + V'z_\eta + Wz_\zeta \end{bmatrix}, \tag{14}$$

where U', V', and W' are the *contravariant* velocity fluctuation components generated in Step 5 and $x_{i,\alpha \Rightarrow \alpha \in \xi,\eta,\zeta}$ are the grid metrics. Once these correlated fields are converted to the physical domain, the known Reynolds stresses are imposed on the sequence generated and the fluctuations can be imposed on the predefined mean velocity and density profiles at the inlet.

An important point to note with such an approach is concerned with curvilinear/axisymmetric grids used in cases such as our jet simulations. Working in the computational domain serves the DF scheme well as we are currently only specifying a *single* integral length scale in each direction. With the grid metrics and the transformation of Eq. 14, the stretching in the grid automatically accounts for larger length scales as one moves away from the wall (or high grid resolution) after conducting the transformation to physical space. However, in the case of axisymmetric grids with a centerline singularity, the azimuthal lengths grow smaller as the grid approaches the singularity. Therefore, the DF will impose unphysical fluctuations near these regions. The approach taken in Dhamankar, *et.al.*[10] is to limit the DF within the boundary layer region and they claim good results when using this approach.

This technique could be implemented into our simulations, if desired. Another approach, first proposed by Veloudis, *et.al.*,[11] is to break the inflow domain into multiple segments, each with their own filter coefficients and length scales. Such an approach would require minimal modifications to the existing routine. Nevertheless, flow geometries such as the flat-plate boundary layer would not suffer from this malady.

B. Additional MODULES

The procedures outlined previously have been implemented in *FDL3DICE* in a series of MODULES. These are:

1. GCOMON_DF
2. DIGFILT
3. MTPRNG

The following sections present some details corresponding to each of the above modules.

1. GCOMON_DF

The GCOMON_DF module was incorporated to include all of the internal variables required by the digital filtering scheme. This includes the variables used to represent the prescribed inflow profiles.

2. DIGFILT

The heart of the digital filtering is located in MODULE DIGFILT. Each function described in §II.G is implemented in its own SUBROUTINE for ease of modification and interpretation. An initialization routine for the digital filtering is also contained in this MODULE. The initialization takes into account reading in all required input files as well as warming up the digital filtering/random number generator.

In terms of the random number generation, the following approach has been adopted for ease of implementation. First, ALL processors read in the inflow boundary layer profile data, regardless of whether they are in inflow processor or not. The expense in terms of additional memory requirements are deemed negligible as we are only concerned with a relatively small two-dimensional domain.

Next, as two independent sequences are required to use the Box-Muller theorem, two independent seeds are generated by the MASTER node. Figure 1 presents an example of the decorrelation between two signals generated by the Mersenne Twister with the independent seeds. In Fig. 1a, the uncorrelated signals are shown, whereas the degree of de-correlation is supported by the low values of the cross-correlation coefficient (Fig. 1b). Once obtained by the MASTER, the seeds are subsequently BROADCAST to all processors. As the entire inflow domain requiring digital filtering is saved in memory in each processor, the random sequences generated by the tool span the *entire* inflow boundary of interest. This is done to maintain consistency between the various processors containing portions of the inflow boundary without having to communicate between processors, which would be more expensive in terms of CPU time. Figure 1c presents a series of 32 signals generated by separate processors after having received the independent seeds from the MASTER node. It is evident that all processors generate the same data, confirming the validity of the approach implemented. After developing the random sequence, each processor simply extracts its portion of the inflow plane from the data.

3. MTPRNG

The Mersenne Twister from Ref. [9] is implemented in MODULE MTPRNG. Since two sequences of random numbers must be generated every time step, the original source was duplicated in order to have independent specifications of the two seeds generated by the MASTER node and also two independent *pseudo*-random sequences. Even though the Box-Muller is applied to decorrelate the random sequences, it was decided to incorporate this logic to further remove any potential correlation between the random numbers. That is, it was deemed preferrable to take this approach than to generate two consecutive sequences with the same seed.

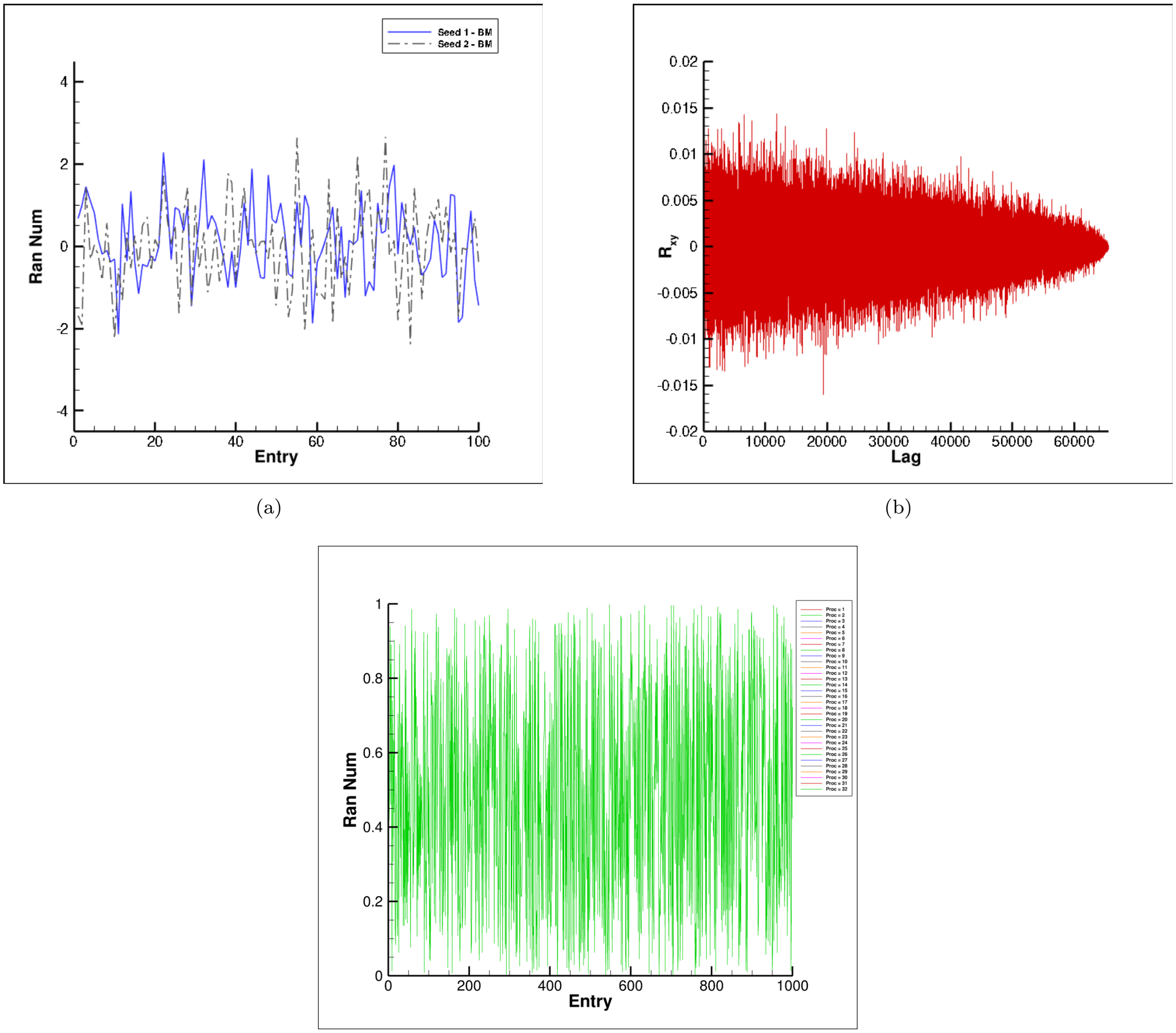
C. Modifications to Existing MODULES

To accomodate the new MODULES, several modifications had to be made in several existing subroutines. In particular, changes were made in:

1. COMDATIN
2. MPIBNDRY

1. COMDATIN Modifications

To ensure backward compatibility in the *FDL3DICE* input deck (*i.e.* the '.dat' file), specification of the digital filtering parameters is made in a separate file passed to the executable upon startup. As all processor indiscriminately read the prescribed inflow profiles, MODULE COMDATIN was modified to search for the DF input file upon execution of the code. If found, the MASTER reads all the inputs and BROADCASTs the data to all processors. The additional logic to search for this file and execute the DF initialization takes place at the end of COMDATIN, once all the case-specific inputs have been read by the MASTER. After reading in the DF input file (to be described in a later section), COMDATIN calls the INITDF subroutine in DIGFILT to initialize all variables required for the procedure.



(a)

(b)

(c)

Figure 1. Example of *Pseudo*-Random Number Sequences Used Throughout the Digital Filtering: (a) Uncorrelated Number Sequence After Executing the Box-Muller Recombination; (b) Cross-Correlation Between the Two Sequences; (c) Sample Run of 32 Processors Developing the Same Sequence with the Broadcast Seeds.

2. Boundary Condition Modifications in MPIBNDRY

Naturally, the boundary conditions must be modified to accomodate the DF. The testing done so far was conducted on a supersonic jet case, which meant the modification of subroutine MPIJETXB. Only the inflow boundary condition in this subroutine had to be modified and very little changes were required. As the DF subroutines have been developed in an object-oriented fashion, the user only needs to provide the code a way of determining whether to execute the DF or not. If DF is to be executed, a call is made to DF_INFLOW *only on the first sub-iteration of each time step.* Therefore, the unsteady profile is kept constant throughout the entire iteration. Once DF_INFLOW is complete, the user can then transfer the unsteady inlet profile to the global primitive variable arrays used in *FDL3DICE*.

With this setup, it should be trivial to extend the use of the DF to cases other than the jet case being studied, particularly if the structure of those cases is already implemented in the code.

D. Additional Files

To execute the DF routine, a series of four files must be provided to *FDL3DICE* upon code startup. These are:

1. Digital Filter Input File – df.dat
2. Digital Filter Input Boundary Grid File – inflow.grd
3. Input Boundary Mean Turbulent Profiles – restress.dat
4. Input Boundary Grid Metrics File – inflow.mtx

Note that the above file names are hard-coded into the initialization routine. A brief synopsis of each file is presented below.

1. Digital Filter Input File – df.dat

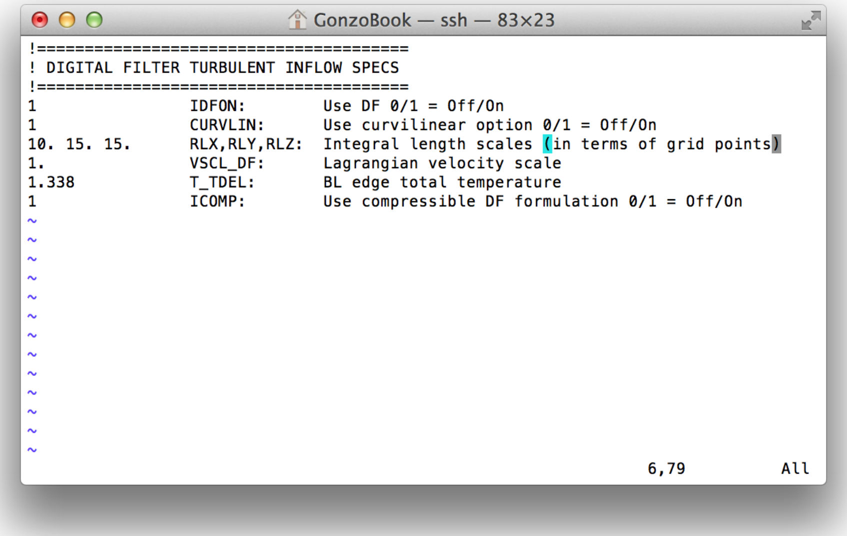
The 'df.dat' file has a very simple structure that is reminiscent of the *FDL3DICE* '.dat' file. Figure 2 shows a screen-capture of the 'df.dat' file. After a series of headers, the user specifies whether or not to use the digital filter based on the value of the 'IDFON' variable. Even though it is highly recommended to conduct the digital filtering in computational space, the user is given the option to select filtering in either the physical or computational domains using the 'CURVLIN' variable. The following three entries correspond to the integral length scales in the x, y, and z directions, respectively, and are specified in terms of *grid points*. In other words, these values are assumed to correspond to the integral scale of interest having already being converted to a number of grid points in the grid being used.

Another variable that is associated with the correlations to be imposed on the random variables is 'VSCL_DF.' It represents the non-dimensional Lagrangian velocity used in the temporal correlation equation, Eq. 8. It is provided here as an option to the user in order to provide a means of specifying a velocity *other* than the reference velocity assumed by the non-dimensionalization of the governing equations. An example of when this may be useful is in the simulation of a jet from stagnation conditions. Here, the flow velocity in the 'stagnation' inflow boundary will have a very small magnitude that is different from, *e.g.*, the sonic conditions in the freestream. The latter are typically assumed as the reference conditions in such simulations. Therefore, having a means of specifying a *local* velocity scale for the inflow plane may be of use in various applications.

Finally, the last two entries in the file are associated with the compressible formulation of the DF, specifically the SRA. First, the user must supply the non-dimansional stagnation temperature at the boundary layer edge for the flow being simulated. This is used in the extended SRA of Cebeci & Smith (Eq.13). Finally, the decision of whether or not to use the compressible formulation is made by the code based on the value of 'ICOMP.'

2. Digital Filter Input Boundary Grid File – inflow.grd

For the inflow plane grid, the user must supply an *unformatted*, two-dimensional Plot-3D grid. The case for which the DF was tested had the inflow plane in the *Y-Z* plane. Because of this, the code was written



```
!=====================================
! DIGITAL FILTER TURBULENT INFLOW SPECS
!=====================================
1                IDFON:        Use DF 0/1 = Off/On
1                CURVLIN:      Use curvilinear option 0/1 = Off/On
10. 15. 15.      RLX,RLY,RLZ:  Integral length scales (in terms of grid points)
1.               VSCL_DF:      Lagrangian velocity scale
1.338            T_TDEL:       BL edge total temperature
1                ICOMP:        Use compressible DF formulation 0/1 = Off/On
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                                6,79          All
```

Figure 2. ***FDL3DICE* Digital Filter Input File Format**

with such variable names. However, one thing to note is that, with the inclusion of the grid metrics file, the input plane grid is currently not used by the algorithm. Therefore, it will most likely be eliminated in the near future. Also, note that this file, as well as the grid metric file, should contain *ONLY* the portion of the inflow plane that requires the specification of unsteady data. The jet flow case is an example of a model that would require a sub-section of the entire inflow plane to specify the nozzle inflow conditions.

3. Input Boundary Mean Turbulent Profiles – restress.dat

The 'restress.dat' file includes both the mean velocity, pressure, and density profiles along with the components of the Reynolds stress tensor. It is an ASCII formatted file that has the following structure:

- File header containing two INTEGER values: NY & NVAR
- 'NVAR' number of columns of data at 'NY' points

NOTE that the input profile is *one-dimensional*. Therefore, 'NY' corresponds to the number of *wall-normal* points in the profile and *MUST* coincide with the first index of the input grid file. If these do not coincide, the code halts execution. For 'NVAR', the user has two options: **7 or 9**. These two options correspond to the fact of having ALL Reynolds stress profiles or ignoring the XZ and YZ components, as they are usually small in comparison to the other components. The structure of the columns is then:

1. Mean Streamwise Velocity, U
2. Mean Pressure, P
3. Mean Density, ρ
4. R_{11} Stress
5. R_{22} Stress
6. R_{33} Stress
7. R_{21} Stress
8. R_{13} Stress, if available
9. R_{23} Stress, if available

4. Input Boundary Grid Metrics File – inflow.mtx

The last file required for the DF to run successfully includes the grid metrics for the inflow plane grid. This is provided in an unformatted, Plot-3D function file containing the following 9 grid metrics on the inflow plane:

1. x_ξ, x_η, x_ζ
2. y_ξ, y_η, y_ζ
3. z_ξ, z_η, z_ζ

The user need only provide this file if the curvilinear option of the DF is invoked. These metrics are then used in Eq. 14 to convert the contravariant velocity fluctuations correlated by the digital filter to Cartesian velocity components.

IV. Sample Inflow Data Sequence

The following section presents some examples of the development of a series of unsteady profiles at the inflow plane of a perfectly-expanded supersonic nozzle. Figure 3 shows a view of the extent of the computational domain adopted in the supersonic jet flow simulations along with a snapshot of the unsteady flow imposed at the nozzle inlet for reference. The following sections will present data generated in this inflow sub-domain to more clearly highlight the performance of the digital filter.

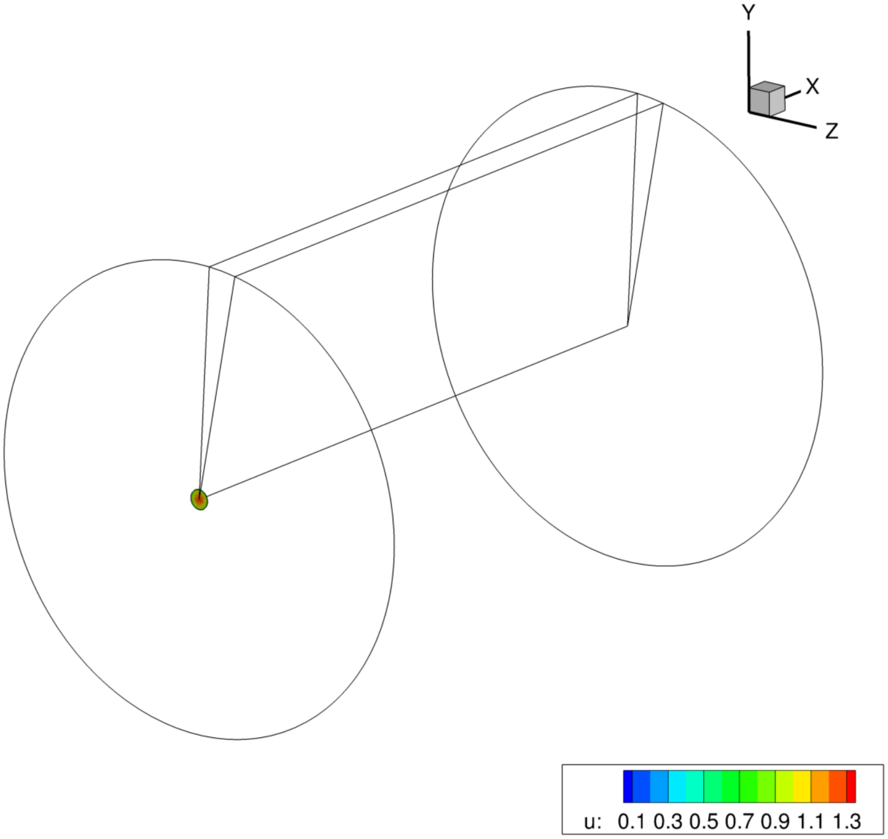


Figure 3. Digital Filter Domain in the Supersonic Nozzle Simulation Case.

A. Comparison of *Physical* vs. *Curvilinear* Implementations

To illustrate the differences inherent in the DF formulations in the physical versus computational domain, Fig. 4 shows a comparison between the profiles generated by the digital filter at a particular point in time. As this is a compressible flow simulation, fluctuations for all three velocity components and the flow density are generated by the digital filter. Recall that pressure is assumed constant in the Strong Reynolds Analogy. Table 1 presents the settings adopted in these tests.

Table 1. Digital Filter Settings in the Supersonic Jet Flow Test Case.

Setting	Value
r_{lx}	10
r_{ly}	15
r_{lz}	15
VSCL_DF	1.0
T_TDEL	1.338

Immediately evident is that, for this axisymmetric case, there is relatively little difference between the two formulations. Again, because the filtering is applied in the entire inflow plane, the reducing size of the azimuthal grid forces the generation of unphysical turbulence near the centerline. A remedy for this would be to cap the filtering at the extent of the boundary layer, as done by Dhamankar, *et.al.*[10] However, this would be dependent on the size of the boundary layer being simulated. In this case, a fully-developed pipe flow velocity profile was assumed at the inlet. Therefore, the boundary layer thickness actually encompasses the entire radius. Additional work needs to be done to evaluate the performance of these two implementations in a non-curvilinear flow problem such as a flat-plate boundary layer.

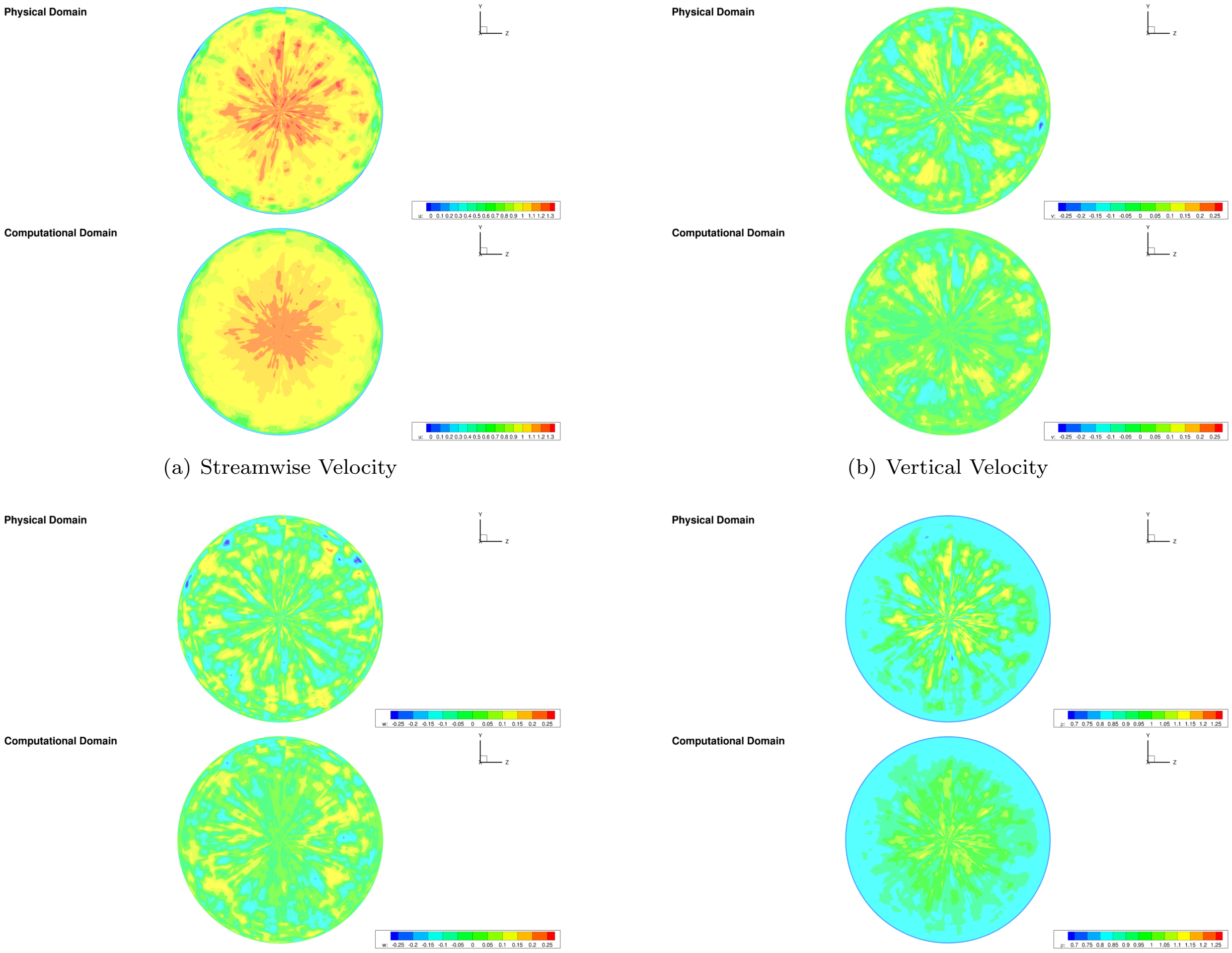
Another fact to point out in the results in Fig. 4 is the apparent azimuthal discontinuity in all of the profiles. This is a figment of the 5-point overlap required by the high-order algorithm. The current DF implementation generates unsteady data at ALL grid points, including the overlap region. Therefore, there will be duplicate entries for coincident points. This is not thought to be detrimental to the code's performance as the overlap should account for an averaging between the two points. Regardless, the DF data is merely a mathematical tool to implement an *approximate* representation of turbulence. As the flow evolves downstream, these discrepancies should be smoothed out by the evolution of the Navier-Stokes equations. In addition, there is nothing in the filtering routine that would guarantee azimuthal periodicity. Therefore, even if the filtering were capped at the final point prior to the overlap, a discontinuity would still be prevalent.

B. Sample Inflow Unsteady Data Sequence

Having verified that the additional MODULEs were properly implemented, the supersonic jet simulation was exercised for a limited number of iterations to evaluate the robustness of the DF implementation. Figure 5 presents several snapshot of the unsteady streamwise velocity generated by the DF using the curvilinear formulation. Note that, for the *bulk-flow* inflow typically used previously, the inflow streamwise velocity would have been set to 1.0. Therefore, the assumed DF parameters result in velocity fluctuations of approximately 30 percent in the streamwise velocity. From the data in Fig. 4, it is evident that similar fluctuation magnitudes in all other variables are generated by the DF in the current case.

V. Closing Remarks

Presented in this report is a brief synopsis of activities being conducted to implement a digital filtering capability within *FDL3DICE*. This report is intended to provide an overview of DF to all parties potentially interested in its use and to briefly describe its implementation. As described above, the DF routine requires minimal user input and can easily be incorporated into any of the standard cases currently available in *FDL3DICE*. To obtain the required inflow grid and metrics, a separate tool was developed in FORTRAN to extract the relevant extent from the overall simulation grid. As for the 'restress.dat' file, the data in this file can be obtained from various external sources. Currently, the data assumed for the jet simulation was



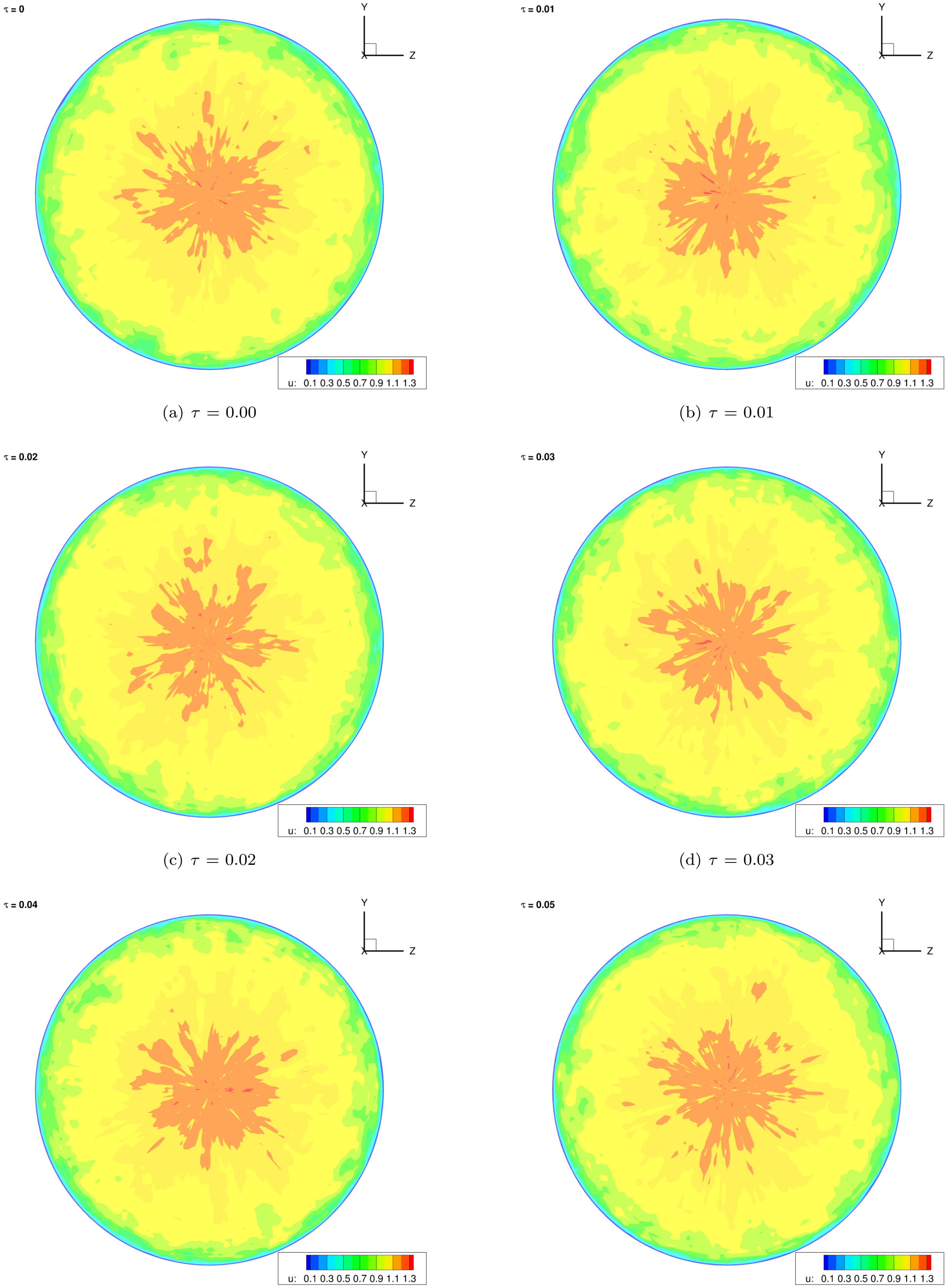
(a) Streamwise Velocity

(b) Vertical Velocity

(c) Lateral Velocity

(d) Density

Figure 4. Comparison of Physical and Curvilinear Digital Filter Formulations.



(a) $\tau = 0.00$

(b) $\tau = 0.01$

(c) $\tau = 0.02$

(d) $\tau = 0.03$

(e) $\tau = 0.04$

(f) $\tau = 0.05$

Figure 5. Inflow Streamwise Velocity Sequence Generated by the Curvilinear Digital Filter.

obtained from available DNS data published online. The extensive simulations conducted at the *HFCMPL* can also provide this data for cases such as the flat-plate boundary layer.

It should be noted that the reason for reading an external file to obtain the inflow plane grid metrics is a matter of ease of implementation. *FDL3DICE* can compute these values organically. However, as each processor only contains portions of the overall mesh, the logic to extract the correct data range would have to be incorporated and the potential for adversely affecting code functionality is too great. In addition, these three-dimensional metrics cannot be computed from the inflow grid provided to the DF since it is two-dimensional. Therefore, providing this data via an input file seemed the most efficient implementation.

The brief results presented in §IV appear to show that the DF has been properly implemented. Potential sources of unphysical turbulence when utilizing this intialization method in axisymmetric grids was briefly described. These sources stem from the reducing size of the grid in the azimuthal direction as the grid nears the centerline discontinuity. Several approaches have been proposed in the literature to address similar issues and these should be relatively simple to incorporate in the future, if so desired. Nevertheless, as the Navier-Stokes equations modulate the flow as it evolves, these issues may not be of great consequence but additional testing is required.

As the current jet grid does not have sufficient grid resolution within the nozzle to support a turbulent flow, modifications to the model need to be made prior to conducting a full simulation with digital filtering as the turbulent inflow method. Future work will be focusing on refining this model and conducting said simulations. The use of this turbulence inflow method will also be investigated in under-expanded jets.

References

[1] Touber, E. and Sandham, N., "Oblique Shock Impinging on a Turbulent Boundary Layer: Low-Frequency Mechanisms," *38th AIAA Fluid Dynamics Conference*, Seattle, WA, 2008, AIAA Paper 2008–4170.

[2] Touber, E. and Sandham, N., "Large-eddy simulation of low-frequency unsteadiness in a turbulent shock-induced separation bubble," *Theoretical and Computational Fluid Dynamics*, Vol. 23, 2009, pp. 79–107.

[3] Xie, Z.-T. and Castro, I., "Efficient Generation of Inflow Conditions for Large Eddy Simulation of Street-Scale Flows," *Flow Turbulence & Combustion*, Vol. 81, 2008, pp. 449–470.

[4] Klein, M., Sadiki, A., and Janicka, J., "A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations," *Journal of Computational Physics*, Vol. 186, 2003, pp. 652–665.

[5] Lund, T., Wu, X., and Squires, K., "Generation of Turbulent Inflow Data for Spatially-Developing Boundary Layer Simulations," *Journal of Computational Physics*, Vol. 140, 1998, pp. 233–258.

[6] Morkovin, M., "Effects of compressibility on turbulent flows," *Mèchanique de la Turbulence*, 1962, pp. 367–380.

[7] Cebeci, T. and Smith, A., *Analysis of turbulent boundary layers*, Academic Press, 1974.

[8] Zhang, Y.-S., Bi, W.-T., Hussain, F., and She, Z.-S., "A generalized Reynolds analogy for compressible wall-bounded turbulent flows," *Journal of Fluid Mechanics*, Vol. 739, 2014, pp. 392–420.

[9] Matsumoto, M. and Nishimura, T., "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," *ACM Transactions on Modeling and Computer Simulation*, Vol. 8, No. 1, 1998, pp. 3–30.

[10] Dhamankar, N., Martha, C., Situ, Y., Aikens, K., Blaisdell, G., Lyrintzis, A., and Li, Z., "Digital Filter-based Turbulent Inflow Generation for Jet Aeroacoustics on Non-Uniform Structured Grids," *52nd AIAA Aerospace Sciences Meeting*, National Harbor, MD, 2014, AIAA Paper 2014-1401.

[11] Veloudis, I., Yang, Z., McGuirk, J., Page, G., and Spencer, A., "Novel Implementation and Assessment of a Digital Filter Based Approach for the Generation of LES Inlet Conditions," *Flow, Turbulence and Combustion*, Vol. 74, No. 1, 2007, pp. 1–24.