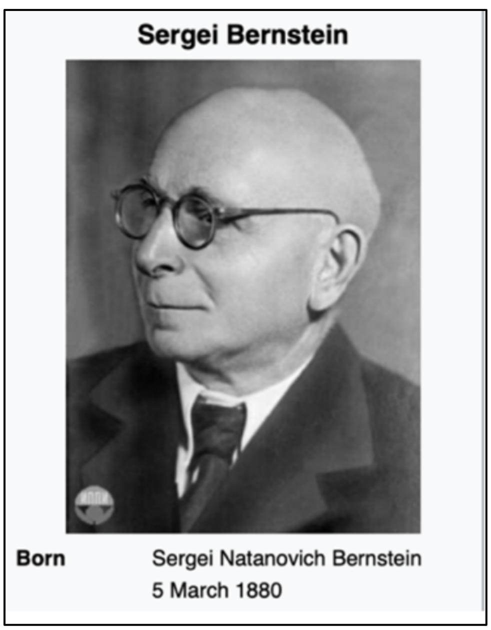
Constrained Field Reconstruction Using Bernstein Polynomial Derived Operators

Aaron H. Auslender, Robin L. Hunt and Louis M. Edelman
NASA Langley Research Center
Hampton, VA 23681-2199

JANNAF
52nd SMBS, 40th APS, 40th EPSS, 34th ESH Subcommittee Meetings
Hilton Salt Lake City Center
Salt Lake City, Utah
December 4-8, 2023

Abstract

This method recasts the mass-conservation equation as a Sylvester equation employing high-order derivative operators derived from modified Bernstein polynomial expansions. Given prescribed velocity fields, the algorithm yields a constrained, least-squared solution for the associated density field. To demonstrate the practical utility of this methodology, it is applied to a computationally-derived two-dimensional isolator dataset. This reconstruction method is envisioned to be used in conjunction with diagnostic techniques to aid in the quantification of isolator flow fields, since obtaining highly characterized datasets within this engine component is exceedingly difficult.



Introduction:

This paper addresses the feasibility of a novel numerical algorithm to directly reconstruct a density field, given a fully prescribed velocity field, via local mass conservation enforcement. This numerical technique is meant to augment experimental diagnostic techniques, since it is very difficult to acquire coupled thermodynamic and fluidic fields in their entirety. Of particular interest to the hypersonic-propulsion design community is the flow-field structure generated within the high-speed isolator component of an engine. Yet, within the isolator section of the hypersonic engines due to volume, weight, environmental constraints and diagnostic-access constraints, only sparse datasets are typically acquired; thereby, inherently limiting the characterization of this environment.

To ameliorate this problem, a reconstruction algorithm is presented in this paper. It is applied to two canonical validation problems (where the fields are analytically obtainable) to verify the basic algorithm. Also, further assessment is performed using a computationally generated flow field solution relevant to a nominal back-pressured isolator environment. The latter case has three significant technical issues. Firstly, the computational derivative operator is not equivalent to the Bernstein-derived derivative operator. Secondly, the grid resolution of the two solutions is not equivalent, since the computational datasets were mapped onto a courser corresponding algorithmic grid. Thirdly, the invoked mapping procedure reduced the geometric volume of the grid which partially obscures the computational boundary conditions. Collectively, these issues influence the reconstructed solution and are discussed in the section of the paper addressing the density-field reconstruction results.

The CFD simulations of a back-pressured isolator serve as a test case for this study. The computational domain reflects the geometry of a small scale direct-connect facility at NASA Langley Research Center, called the Isolator Dynamics Research Laboratory [1], and spans the facility plenum, Mach 2.5 nozzle and entire isolator duct. Given the geometrical symmetry of this flow path and the fact that only steady-state simulations are performed, a structured grid of one-quarter of the flow path cross-section is used. Grid points are clustered near the walls such that the y+ value at any cell center adjacent to the surface is acceptably small. The grid points are also clustered in the streamwise direction at the nozzle throat, with the spacing gradually increasing towards the nozzle exit and fixed throughout the isolator. CFD solutions are found using the VULCAN (Viscous Upwind aLgorithm for Complex flow ANalysis) Navier-Stokes flow solver [2]. Results are obtained by numerically solving the Reynolds-averaged Navier-Stokes equations until steady-state conditions are achieved. The Menter baseline turbulence model [3] was chosen for this study and the boundary condition imposed along all solid surfaces is an adiabatic no-slip condition. For the CFD solution presented, the subsonic plenum total pressure and total temperature are specified as 844,136 Pa and 292.82 K. Also, the isolator exit pressure is specified as 305,067 Pa, which consequently generates a shock-train structure within the isolator section. This computation yielded the centerline density and velocity fields, subsequently displayed in Figures 1-3.

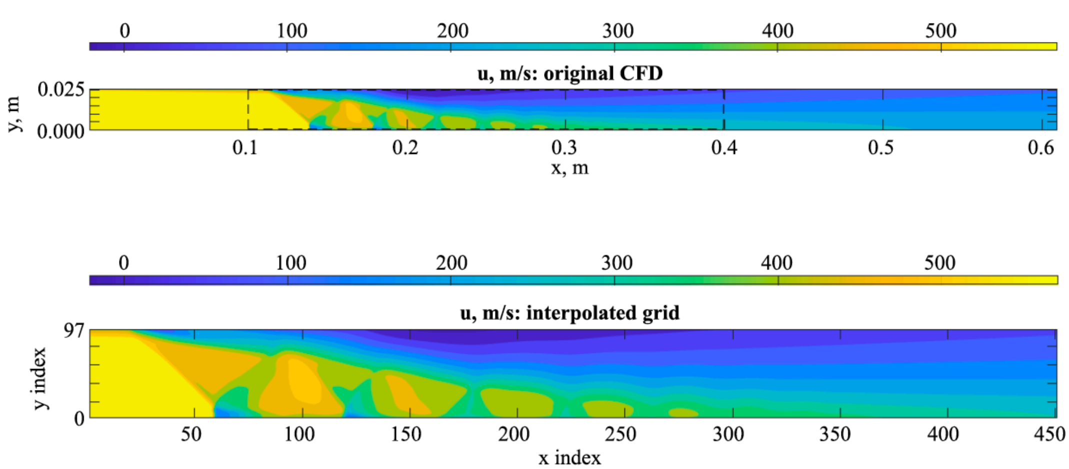


Figure 1: Isolator Dynamics Research Laboratory Computationally Derived U-Velocity Field.

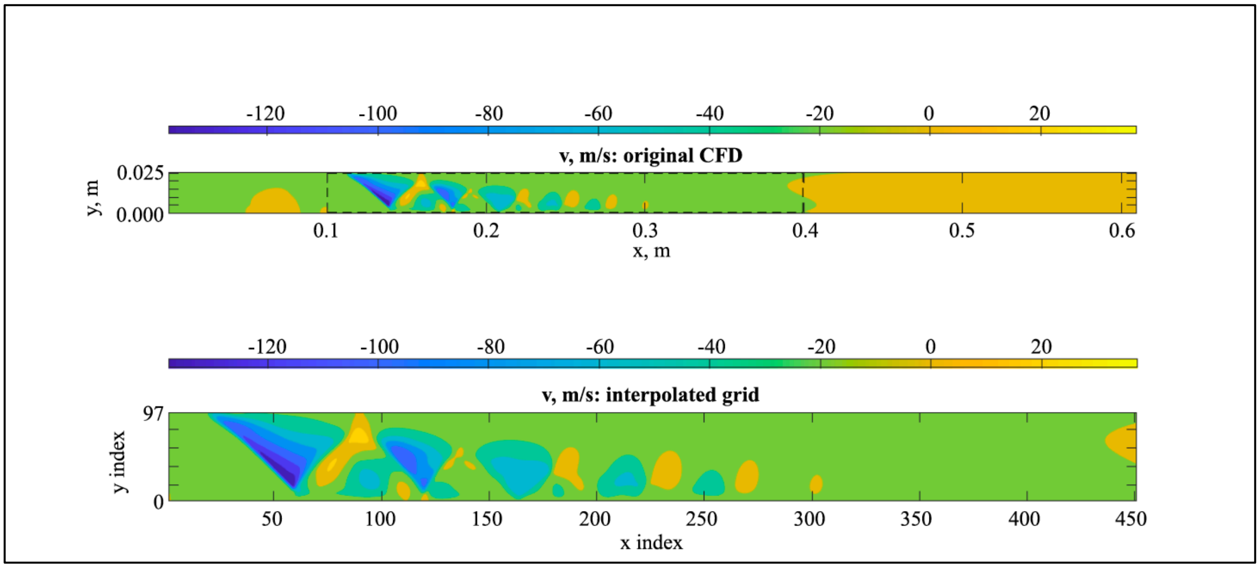


Figure 2: Isolator Dynamics Research Laboratory Computationally Derived V-Velocity Field.

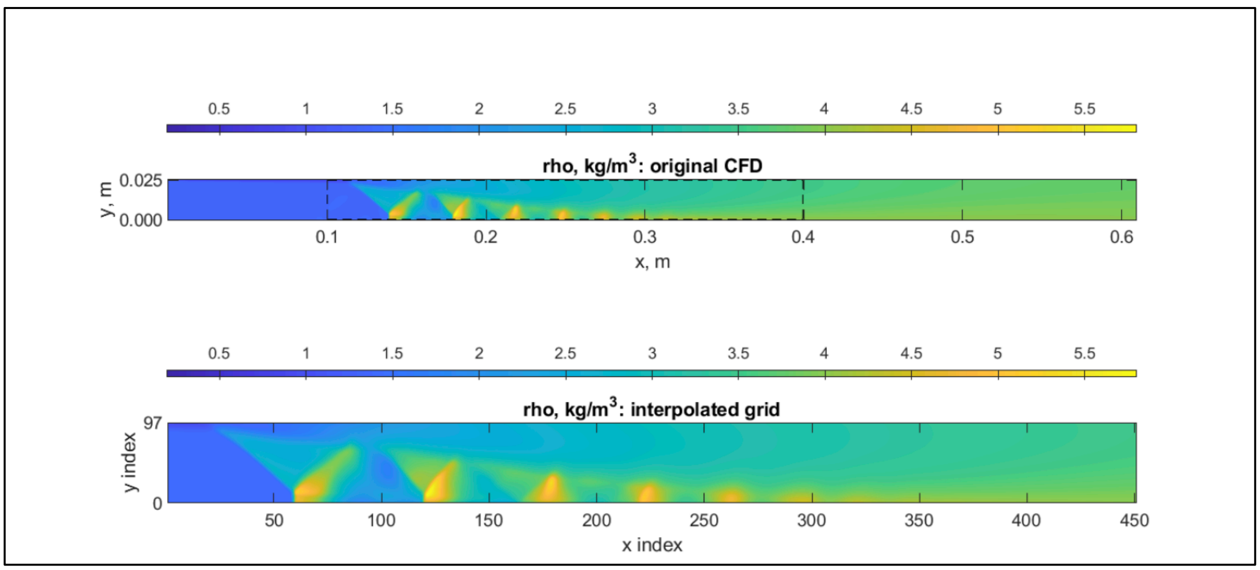


Figure 3: Isolator Dynamics Research Laboratory Computationally Derived Density Field.

Solution Methodology:

The goal of this study is to reconstruct the two-dimensional density field corresponding to a prescribed two-dimensional set of velocity measurements, without invoking any explicit smoothing and/or filtering of the velocity measurements. The latter constraint simplifies the reconstruction process, since no a priori knowledge is required and/or assumed concerning the noise and/or errors inherent within the given velocity fields. For this study, a Bernstein polynomial expansion is employed since this nonorthogonal basis-set decomposition is well suited to dampen oscillatory behavior, albeit at the detriment of the convergence rate. Equations 1-3 detail the mathematical formulation first proposed by Sergei Bernstein in 1912 [4], in which discretized-function data are used as weighting values to construct an approximating polynomial. Note that Bernstein polynomials have attributes of positivity, continuity, and form a partition of unity over a closed interval. Hence, Bernstein polynomials are highly compatible with the construction of derivative operators, a feature that is exploited by this numerical reconstruction technique.

(1) $$\mathfrak{I}_B(x) \equiv \sum_{i=0}^{n} F\left(a + \left(\frac{i}{n}\right)(b-a)\right)\binom{n}{i}\left(\frac{x-a}{b-a}\right)^{(i)}\left(\frac{b-x}{b-a}\right)^{(n-i)} \equiv \sum_{i=0}^{n} F_{(i,n)}\, B_i^n(x)$$

(2) $$a \leq x \leq b$$

(3) $$n = \frac{(b-a)}{\Delta x}$$

Having constructed the Bernstein polynomial expansion of a function, various relationships involving the derivative operator are available, as shown in Equation 4 and Equation 5. Note that the latter part of Equation 6 [5] specifies an equivalency amongst weighting values associated with derivatives of the Bernstein polynomial expansion and weighting values of the non-transformed function. This is a very useful relationship, since upon further manipulation, Equation 7 is obtained, which directly relates transformed derivative operators and nontransformed derivative operators.

(4) $$\mathfrak{I}_B^{(q)}(x) = \frac{d^q}{dx^q}\sum_{i=0}^{n} F_{(i,n)}\, B_i^n(x) = \sum_{i=0}^{n} F_{(i,n)}\frac{d^q}{dx^q}B_i^n(x) = \sum_{i=0}^{n} a_{(i,n)}^{(q)}\, B_i^n(x)$$

(5) $$\mathfrak{I}_B^{(q)}(x) = \sum_{i=0}^{n} a_{(i,n)}^{(q)}\, B_i^n(x) = \sum_{i=0}^{n} F_{(i,n)}^{(q)}\, B_i^n(x) \approx \sum_{i=0}^{n} \mathfrak{I}_{B}{}_{(i,n)}^{(q)}\, B_i^n(x)$$

(6) $$\mathfrak{I}_{B}{}_{(i,n)}^{(q)} \approx a_{(i,n)}^{(q)} = \sum_{k=-q}^{q}\sum_{m=0}^{q}\frac{q!\,(-1)^{(m+q)}}{(n\,\Delta x)^q}\binom{q}{m}\binom{i}{m+k}\binom{n-i}{q-(m+k)}\, F_{(i-k,n)}$$

(7) $$\mathfrak{I}_{B}{}_{(i,n)}^{(q)} \approx \sum_{k=-q}^{q}\sum_{m=0}^{q}\frac{q!\;(-1)^{(m+q)}}{(n\,\Delta x)^q}\binom{q}{m}\binom{i}{m+k}\binom{n-i}{q-(m+k)}\sum_{s=0}^{N\to\infty}\frac{(-k)^{(s)}\,(\Delta x)^{(s)}}{s!}\, F_{(i,n)}^{(s)}$$

Upon truncation of Equation 7, a square invertible matrix is constructed and used to generate a linear equation set. For this study, a fifteen-by-fifteen dimensioned matrix is derived and is symbolically presented in Equation 8. Note that approximations to the exact spatial derivative operators are explicitly available via a matrix inversion. The reader is directed to the Appendix for a partial listing of these elements. This information is the crux of the algorithm. Additionally, other non-square analytic formulations are possible, but are not explored in this study and are left for future efforts.

(8) $$\begin{pmatrix} \mathfrak{I}_{B}{}_{(i,n)}^{(0)} \\ \mathfrak{I}_{B}{}_{(i,n)}^{(1)} \\ \vdots \\ \mathfrak{I}_{B}{}_{(i,n)}^{(N)} \end{pmatrix} \approx \left[\Pi_{(i,n)}^{(N,\Delta x)}\right] \begin{pmatrix} F_{(i,n)}^{(0)} \\ F_{(i,n)}^{(1)} \\ \vdots \\ F_{(i,n)}^{(N)} \end{pmatrix}$$

Combining these equations yields various stencils (weighted combinations of adjacent discretized function values) that approximate the exact spatial derivative operators. These forms are symbolically presented in Equation 9. Note that since this analytic technique does not utilize information exterior to the domain, it is straightforward to implement at (and near) the boundary of the domain, independent of the chosen-truncation value. This property is a direct consequence of the Bernstein polynomial expansion formulation.

$$(9)\quad \left[D^{(\mathrm{B})}_{(\mathrm{x})}\right] = \begin{bmatrix} \Omega^{(0)}_{(\mathrm{i=0,n})} & \cdots & \Omega^{(0)}_{(\mathrm{i=n,n})} \\ \vdots & \ddots & \vdots \\ \Omega^{(\mathrm{n})}_{(\mathrm{i=0,n})} & \cdots & \Omega^{(\mathrm{n})}_{(\mathrm{i=n,n})} \end{bmatrix}$$

Applying the discretized spatial derivative operator to the mass-conservation equation (Equation 10) results in a Sylvester equation (Equation 11). Upon further manipulation, a set of linear equations are generated, consistent with the prescribed velocity fields. Specifically, this is accomplished by invoking the principle of superposition for linear systems which allows the Sylvester equation to be recast. This procedure results in a density response matrix corresponding to each grid point. Subsequently, these matrices are reshaped into a set of column vectors and are subsequently partitioned between the boundary and interior domain grid points yielding Equation 12. Note that this representation explicitly incorporates the boundary conditions, denoted as ρ_{L}, and interior domain values, denoted as ρ_{R}. In this study, a constrained, least-squares formulation of Equation 12 yields Equation 13, with the μ_1 parameter controlling the transverse curvature of the solution (via the restriction of the third derivative), and the μ_2 parameter controlling the condition number of the matrix [6]. The μ_1 parameter constrains the reconstruction to prevent overfitting in the transverse direction, whereas the μ_2 parameter is required due to the ill-conditioned nature of the $[A]$ matrix. Note that the loss of the information associated with the reduced number of boundary-condition points near the wall and centerline, resulting from the mapping procedure, is partially ameliorated by the constraint equation, designed to restrict the reconstruction from obtaining, along the horizontal boundaries of the reconstructed field, values less than a user-specified threshold $\mathrm{d}_{\mathrm{min}}$.

$$(10)\quad \frac{\partial(\rho \mathrm{u})}{\partial \mathrm{x}} + \frac{\partial(\rho \mathrm{v})}{\partial \mathrm{y}} = 0$$

$$(11)\quad \left[[X_{(\rho_{\mathrm{L}},\rho_{\mathrm{R}})}] \circ [U_{\mathrm{CFD}}]\right]\left[D^{(\mathrm{B})}_{(\mathrm{x})}\right] + [D^{(\mathrm{B})}_{(\mathrm{y})}]^{\mathrm{T}} \left[[X_{(\rho_{\mathrm{L}},\rho_{\mathrm{R}})}] \circ [V_{\mathrm{CFD}}]\right] = 0$$

$$(12)\quad [M_{\mathrm{L}}]\, \rho_{\mathrm{L}} + [M_{\mathrm{R}}]\, \rho_{\mathrm{R}} = 0$$

$$(13)\quad \left\| \begin{pmatrix} [M_{\mathrm{R}}] \\ \mu_1 \left[[D^{(\mathrm{B})}_{(\mathrm{y})}]^{\mathrm{T}} [D^{(\mathrm{B})}_{(\mathrm{y})}]^{\mathrm{T}} [D^{(\mathrm{B})}_{(\mathrm{y})}]^{\mathrm{T}}\right] \\ \mu_2\, [\mathrm{I}] \end{pmatrix} \rho_{\mathrm{R}} - \begin{pmatrix} [-M_{\mathrm{L}}]\, \rho_{\mathrm{L}} \\ 0 \\ 0 \end{pmatrix} \right\|_2^2 \equiv \| [A]\, \rho_{\mathrm{R}} - \mathrm{b} \|_2^2 \ \text{ st } \ [C]\, \rho_{\mathrm{R}} = \mathrm{d}_{\mathrm{min}}$$

The solution is obtained by solving the constrained, least-square problem (Equation-13) in a piecewise manner, partitioned in the axial direction. Primarily, this is done to form numerically tractable zones. Note that obtaining an inverse of $[A^{\mathrm{T}}A]$ only requires a symmetric SVD decomposition (Equation-14). In this study, the symmetric SVD decomposition is generated using sequential unitary rotation operations [7]. Also, note that due to the large condition numbers associated with the matrix characterizing this problem, 128-bit quadruple precision is utilized in the FORTRAN coding of the algorithm. Lastly, the closed form solution of Equation 13 that incorporates the constraint function is invoked to yield the reconstructed density field (Equation 15).

$$(14)\quad [A^{\mathrm{T}}A]^{-1} = \left[[Q]\ [\Sigma^{+}]^{-1}\ [Q]^{\mathrm{T}}\right]_{\mathrm{SVD}}$$

$$(15)\quad (\hat{\rho}_{\mathrm{R}})_{(\mu_1,\mu_2)} = [A^{\mathrm{T}}A]^{-1} (A^{\mathrm{T}}\,\mathrm{b}) + [[A^{\mathrm{T}}A]^{-1} [C]^{\mathrm{T}}]\ \ [[C]\ [A^{\mathrm{T}}A]^{-1} [C]^{\mathrm{T}}]^{-1} \left(\mathrm{d} - [C]\, [A^{\mathrm{T}}A]^{-1} (A^{\mathrm{T}}\,\mathrm{b})\right)$$

Canonical Mach-2 Normal-Shock Case Results:

To verify that the algorithm is properly implemented, two canonical cases are presented. Each is an inviscid, Mach-2, planar normal-shock case, only differing in their coordinate system orientation. Specifically, the alignment between the planar normal-shock front and the axial grid coordinate is either 90 degrees or 45 degrees and is denoted as grid-aligned and non-grid-aligned, respectively. For these cases the solutions are derived using a fourth-order truncation scheme (N equals 4), and therefore, at any given grid-point location, a maximum of 17 grid-points are active in the two-dimensional stencil.

Both test cases correspond to a Mach-2 normal shock with a constant specific-heat ratio of 1.40, a value consistent with diatomic molecules (at modest temperatures and pressures). For reference, a Mach-2 normal shock generates a density ratio of 2.667, as do the reconstructed solutions. The grid-aligned reconstructed solution uses the velocity field presented in Figure 4 (left-hand side), as well as specified (and compatible) upstream density boundary conditions. Furthermore, invoking that total-enthalpy is constant throughout the flow field enables a complete reconstruction of the thermodynamic flow field. These data are also shown in Figure 4 (right-hand side). The subsequent non-grid-aligned case is almost identical, with the only modification being that the prescribed velocity field and boundary conditions correspond to a 45-degree rotation of the planar normal shock front, as well as an arbitrarily positioned y-axis intercept location. The results of this calculation are presented in Figure 5. As desired, these data are also consistent with the normal shock "jump" conditions. Note that the reconstructed density fields satisfy Equation 15, with both the μ_1 parameter and the μ_2 parameter set equal to zero and the constraint equation deactivated.

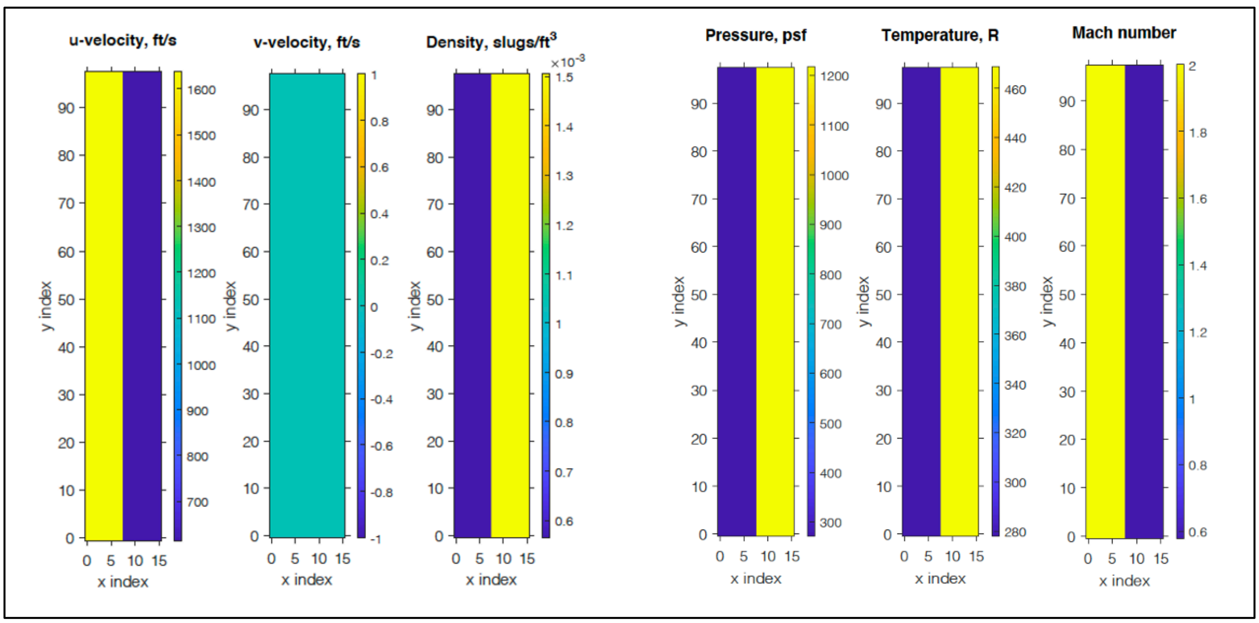


Figure 4: Grid-Aligned Mach-2 Normal Shock Test Case Results (Fourth-Order Truncation)

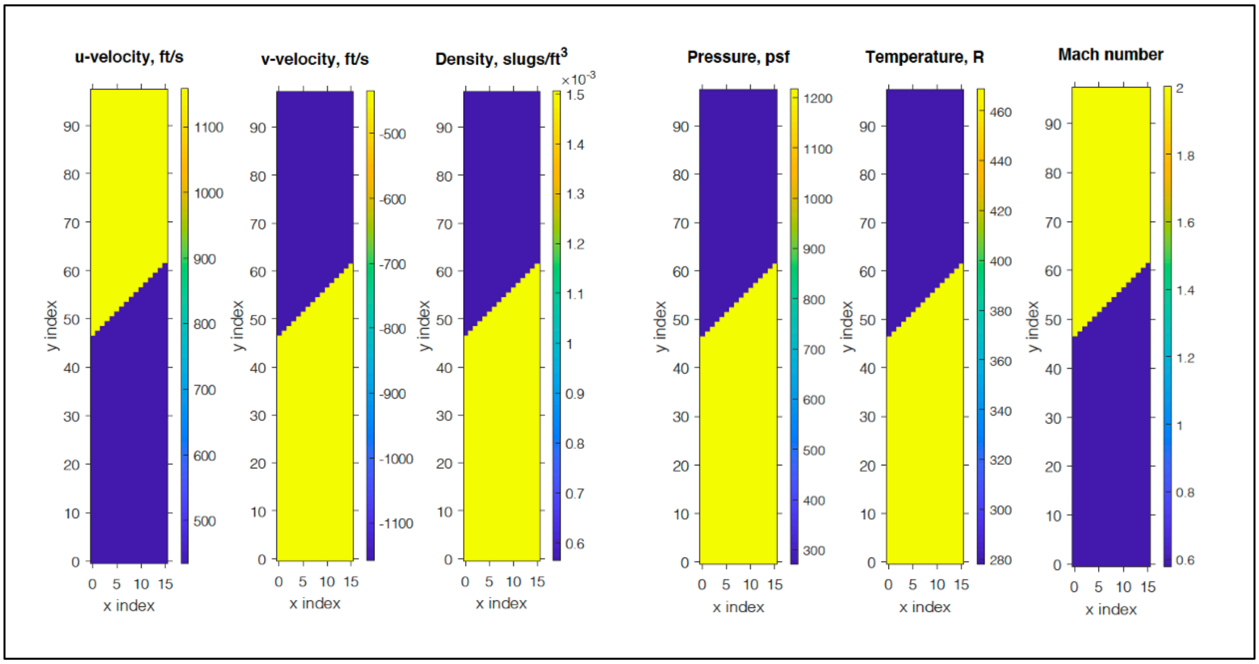


Figure 5: Non-Grid-Aligned Mach-2 Normal Shock Test Case Results (Fourth-Order Truncation)

Comparison of CFD versus Reconstructed Density Fields and Conclusions:

Applying the reconstruction algorithm in a piecewise-axial manner, the concatenated reconstructed density field is obtained (Figure-6). Note that throughout the reconstruction process the μ_1 parameter was set equal to 10^{-3}, the μ_2 parameter was set equal to 10^{-3} times the maximum value of $\sqrt{[\Sigma^+]}$, and the d_{min} value was set equal to a value of $1.940^{-1}\mathrm{kg/m^3}$. These values were empirically derived to yield a reasonable reconstruction.

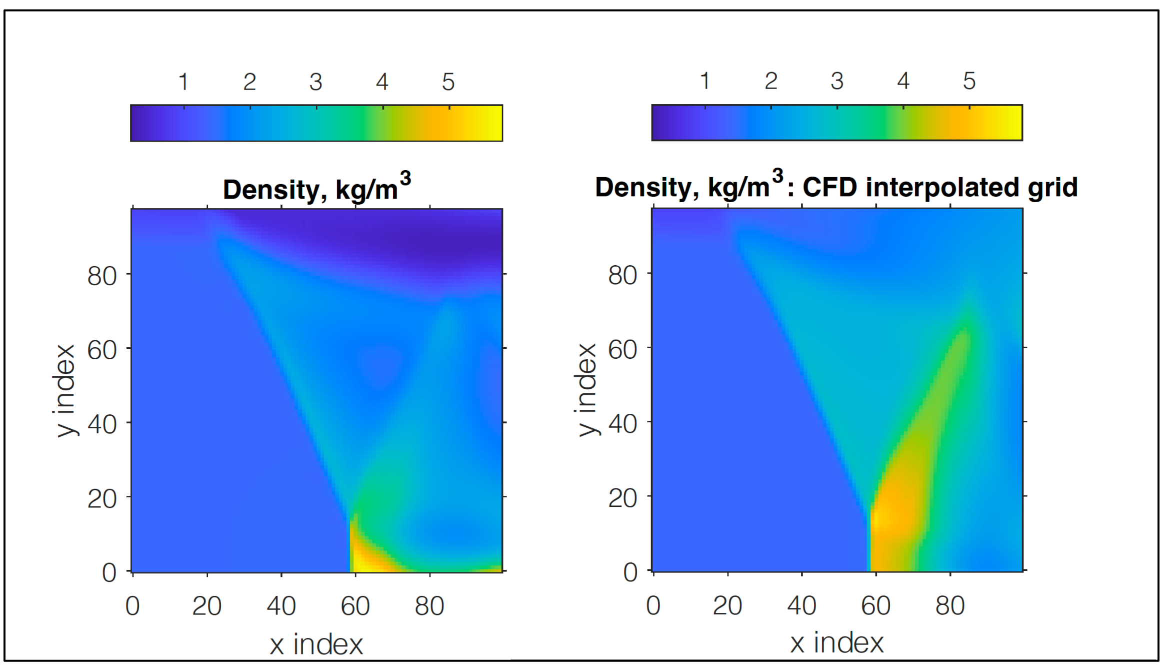


Figure 6: Reconstructed Isolator Dynamics Research Laboratory Density Field vs Computational Data

The derived density field is engineeringly acceptable but lacks a rigorous “match” to the CFD density field; however, this observation needs to be caveated by the complicating issue that the algorithm derivative operator used in this study (N=2) and the CFD derivative operator are not equivalent. Hence, the solutions will not be equivalent. Also, it is interesting to note that the two solutions are not derived using the same spatial resolution, with the CFD being significantly more resolved prior to the mapping procedure (to the courser grid). An additional consequence of the mapping procedure is the loss of information associated with the reduction in boundary condition points (along the centerline and the solid surfaces). This effectively alters the span to the column space of the $[A]$ matrix, which ultimately dictates which solutions are obtainable. None of these influences were explicitly characterized in this study.

In short, the metrics for comparisons need to be examined more thoroughly, but it is reasonable to assume that the as-proposed algorithm needs a modified set of constraint functions to ultimately yield more compatible results, or perhaps a more radical reformulation is required to address the shortcomings. One potentially germane option is to investigate the “splitting” of the $[A]$ matrix into a low-rank matrix and a sparse matrix to access a more suitable inverse problem. Another alternative is to perform the minimization using alternative matrix norms. Clearly, numerous options need to be explored if higher fidelity reconstructions are to be obtained in a reliable manner.

Appendix: Ten Elements of the Inverted Derivative Transform Matrix Operator Pertaining to the Approximation to the Exact First-Order Derivative Operator Used in this Reconstruction Method

These terms are used in conjunction with Bernstein polynomial derivative operators to construct an approximation to the exact first-order derivative operator, see Equation 8. These terms are dependent on the spatial location, the order of the Bernstein polynomial expansion, and the grid spacing interval. Note that a symbolic mathematical computation program, Mathematica [15], was used to aid in the manipulation of the equations and terms of interest.

(FORTRAN VERSION)

$$\left[\, \Pi_{\mathbf{(i,n)}}^{\mathbf{(N,\Delta x)}} \,\right]^{-1}$$

```
 MATRIX_QQ_DIAGONALS( 1, 1,I) = 1.00Q+0000
 MATRIX_QQ_DIAGONALS( 1, 2,I) =(5.00Q-0001*(2.00Q+0000*I_REAL-1.00Q
&+0000*N_REAL)*DX)/(-1.00Q+0000+N_REAL)
 MATRIX_QQ_DIAGONALS( 1, 3,I) =(3.33333333333333333333333333333333Q-
&0001*(6.00Q+0000*I_REAL**2.00Q+0000-6.00Q+0000*I_REAL*N_REAL+N_REA
&L**2.00Q+0000)*DX**2.00Q+0000)/(2.00Q+0000-3.00Q+0000*N_REAL+N_REA
&L**2.00Q+0000)
 MATRIX_QQ_DIAGONALS( 1, 4,I) =(2.50Q-0001*(2.00Q+0000*I_REAL-1.00Q
&+0000*N_REAL)*((-1.00Q+0000+N_REAL)*N_REAL**2.00Q+0000+2.00Q+0000*
&I_REAL**2.00Q+0000*(-6.00Q+0000+5.00Q+0000*N_REAL)-2.00Q+0000*I_RE
&AL*N_REAL*(-6.00Q+0000+5.00Q+0000*N_REAL))*DX**3.00Q+0000)/((-3.00
&Q+0000+N_REAL)*(-2.00Q+0000+N_REAL)*(-1.00Q+0000+N_REAL)**2.00Q+00
&00)
 MATRIX_QQ_DIAGONALS( 1, 5,I) =(2.00Q-0001*(1.00Q+0001*I_REAL*(3.00
&Q+0000-2.00Q+0000*N_REAL)*N_REAL**3.00Q+0000+(-1.00Q+0000+N_REAL)*
&N_REAL**4.00Q+0000+3.00Q+0001*I_REAL**2.00Q+0000*N_REAL**2.00Q+000
&0*(-5.00Q+0000+3.00Q+0000*N_REAL)+1.00Q+0001*I_REAL**4.00Q+0000*(-
&1.20Q+0001+7.00Q+0000*N_REAL)-2.00Q+0001*I_REAL**3.00Q+0000*N_REAL
&*(-1.20Q+0001+7.00Q+0000*N_REAL))*DX**4.00Q+0000)/((-4.00Q+0000+N_
&REAL)*(-3.00Q+0000+N_REAL)*(-2.00Q+0000+N_REAL)*(-1.00Q+0000+N_REA
&L)**2.00Q+0000)
 MATRIX_QQ_DIAGONALS( 1, 6,I) =(1.66666666666666666666666666666667Q-
&0001*(2.00Q+0000*I_REAL-1.00Q+0000*N_REAL)*((-2.00Q+0000+N_REAL)*(
&-1.00Q+0000+N_REAL)**2.00Q+0000*N_REAL**4.00Q+0000-2.00Q+0000*I_RE
&AL*N_REAL**3.00Q+0000*(-6.00Q+0001+1.17Q+0002*N_REAL-7.10Q+0001*N_
&REAL**2.00Q+0000+1.40Q+0001*N_REAL**3.00Q+0000)+6.00Q+0000*I_REAL*
&*4.00Q+0000*(-1.20Q+0002+2.14Q+0002*N_REAL-1.18Q+0002*N_REAL**2.00
&Q+0000+2.10Q+0001*N_REAL**3.00Q+0000)-1.20Q+0001*I_REAL**3.00Q+000
&0*N_REAL*(-1.20Q+0002+2.14Q+0002*N_REAL-1.18Q+0002*N_REAL**2.00Q+0
&000+2.10Q+0001*N_REAL**3.00Q+0000)+2.00Q+0000*I_REAL**2.00Q+0000*N
&_REAL**2.00Q+0000*(-4.20Q+0002+7.59Q+0002*N_REAL-4.25Q+0002*N_REAL
&**2.00Q+0000+7.70Q+0001*N_REAL**3.00Q+0000))*DX**5.00Q+0000)/((-5.
&00Q+0000+N_REAL)*(-4.00Q+0000+N_REAL)*(-3.00Q+0000+N_REAL)*(-2.00Q
&+0000+N_REAL)**2.00Q+0000*(-1.00Q+0000+N_REAL)**3.00Q+0000)
 MATRIX_QQ_DIAGONALS( 1, 7,I) =(1.42857142857142857142857142857143Q-
&0001*((-2.00Q+0000+N_REAL)*(-1.00Q+0000+N_REAL)**2.00Q+0000*N_REAL
&**6.00Q+0000-1.40Q+0001*I_REAL*N_REAL**5.00Q+0000*(-1.80Q+0001+3.2
&0Q+0001*N_REAL-1.70Q+0001*N_REAL**2.00Q+0000+3.00Q+0000*N_REAL**3.
&00Q+0000)+4.20Q+0001*I_REAL**2.00Q+0000*N_REAL**4.00Q+0000*(-8.60Q
&+0001+1.35Q+0002*N_REAL-6.40Q+0001*N_REAL**2.00Q+0000+1.00Q+0001*N
&_REAL**3.00Q+0000)-1.40Q+0002*I_REAL**3.00Q+0000*N_REAL**3.00Q+000
&0*(-1.20Q+0002+1.79Q+0002*N_REAL-8.10Q+0001*N_REAL**2.00Q+0000+1.2
&0Q+0001*N_REAL**3.00Q+0000)+2.80Q+0001*I_REAL**6.00Q+0000*(-3.60Q+
&0002+5.22Q+0002*N_REAL-2.30Q+0002*N_REAL**2.00Q+0000+3.30Q+0001*N_
&REAL**3.00Q+0000)-8.40Q+0001*I_REAL**5.00Q+0000*N_REAL*(-3.60Q+000
&2+5.22Q+0002*N_REAL-2.30Q+0002*N_REAL**2.00Q+0000+3.30Q+0001*N_REA
&L**3.00Q+0000)+7.00Q+0001*I_REAL**4.00Q+0000*N_REAL**2.00Q+0000*(-
&4.80Q+0002+7.01Q+0002*N_REAL-3.11Q+0002*N_REAL**2.00Q+0000+4.50Q+0
&001*N_REAL**3.00Q+0000))*DX**6.00Q+0000)/((-6.00Q+0000+N_REAL)*(-5
&.00Q+0000+N_REAL)*(-4.00Q+0000+N_REAL)*(-3.00Q+0000+N_REAL)*(-2.00
&Q+0000+N_REAL)**2.00Q+0000*(-1.00Q+0000+N_REAL)**3.00Q+0000)
 MATRIX_QQ_DIAGONALS( 1, 8,I) =(1.25Q-0001*(2.00Q+0000*I_REAL-1.00Q
&+0000*N_REAL)*((-1.00Q+0000+N_REAL)**3.00Q+0000*N_REAL**6.00Q+0000
&*(6.00Q+0000-5.00Q+0000*N_REAL+N_REAL**2.00Q+0000)-2.00Q+0000*I_RE
&AL*(-1.00Q+0000+N_REAL)**2.00Q+0000*N_REAL**5.00Q+0000*(-7.56Q+000
&2+7.03Q+0002*N_REAL-2.32Q+0002*N_REAL**2.00Q+0000+2.70Q+0001*N_REA
&L**3.00Q+0000)+2.00Q+0000*I_REAL**2.00Q+0000*N_REAL**4.00Q+0000*(-
&1.5876Q+0004+4.1419Q+0004*N_REAL-3.99Q+0004*N_REAL**2.00Q+0000+1.7
&871Q+0004*N_REAL**3.00Q+0000-3.838Q+0003*N_REAL**4.00Q+0000+3.24Q+
&0002*N_REAL**5.00Q+0000)+4.00Q+0000*I_REAL**6.00Q+0000*(-3.024Q+00
&04+7.3368Q+0004*N_REAL-6.5724Q+0004*N_REAL**2.00Q+0000+2.7482Q+000
&4*N_REAL**3.00Q+0000-5.495Q+0003*N_REAL**4.00Q+0000+4.29Q+0002*N_R
&EAL**5.00Q+0000)-1.20Q+0001*I_REAL**5.00Q+0000*N_REAL*(-3.024Q+000
&4+7.3368Q+0004*N_REAL-6.5724Q+0004*N_REAL**2.00Q+0000+2.7482Q+0004
&*N_REAL**3.00Q+0000-5.495Q+0003*N_REAL**4.00Q+0000+4.29Q+0002*N_RE
&AL**5.00Q+0000)-4.00Q+0000*I_REAL**3.00Q+0000*N_REAL**3.00Q+0000*(
&-4.536Q+0004+1.12572Q+0005*N_REAL-1.0323Q+0005*N_REAL**2.00Q+0000+
&4.4159Q+0004*N_REAL**3.00Q+0000-9.047Q+0003*N_REAL**4.00Q+0000+7.2
&6Q+0002*N_REAL**5.00Q+0000)+6.00Q+0000*I_REAL**4.00Q+0000*N_REAL**
&2.00Q+0000*(-6.552Q+0004+1.59804Q+0005*N_REAL-1.4395Q+0005*N_REAL*
&*2.00Q+0000+6.0523Q+0004*N_REAL**3.00Q+0000-1.2174Q+0004*N_REAL**4
&.00Q+0000+9.57Q+0002*N_REAL**5.00Q+0000))*DX**7.00Q+0000)/((-7.00Q
&+0000+N_REAL)*(-6.00Q+0000+N_REAL)*(-5.00Q+0000+N_REAL)*(-4.00Q+00
&00+N_REAL)*(-3.00Q+0000+N_REAL)**2.00Q+0000*(-2.00Q+0000+N_REAL)**
&2.00Q+0000*(-1.00Q+0000+N_REAL)**4.00Q+0000)
 MATRIX_QQ_DIAGONALS( 1, 9,I) =(1.11111111111111111111111111111111Q-
&0001*((-3.00Q+0000+N_REAL)*(-2.00Q+0000+N_REAL)**2.00Q+0000*(-1.00
&Q+0000+N_REAL)**3.00Q+0000*N_REAL**8.00Q+0000-1.20Q+0001*I_REAL*(-
&1.00Q+0000+N_REAL)**2.00Q+0000*N_REAL**7.00Q+0000*(5.10Q+0002-6.53
&Q+0002*N_REAL+3.13Q+0002*N_REAL**2.00Q+0000-6.90Q+0001*N_REAL**3.0
&0Q+0000+6.00Q+0000*N_REAL**4.00Q+0000)+6.00Q+0000*I_REAL**2.00Q+00
&00*N_REAL**6.00Q+0000*(3.63Q+0004-1.04524Q+0005*N_REAL+1.173850Q+00
&05*N_REAL**2.00Q+0000-6.6342Q+0004*N_REAL**3.00Q+0000+2.0163Q+0004
&*N_REAL**4.00Q+0000-3.192Q+0003*N_REAL**5.00Q+0000+2.10Q+0002*N_RE
&AL**6.00Q+0000)-2.52Q+0002*I_REAL**5.00Q+0000*N_REAL**3.00Q+0000*(
&9.072Q+0004-2.33964Q+0005*N_REAL+2.37204Q+0005*N_REAL**2.00Q+0000-
&1.21916Q+0005*N_REAL**3.00Q+0000+3.3779Q+0004*N_REAL**4.00Q+0000-4
&.849Q+0003*N_REAL**5.00Q+0000+2.86Q+0002*N_REAL**6.00Q+0000)+1.80Q
&+0001*I_REAL**8.00Q+0000*(2.4192Q+0005-6.17184Q+0005*N_REAL+6.1932
&Q+0005*N_REAL**2.00Q+0000-3.15172Q+0005*N_REAL**3.00Q+0000+8.6458Q
&+0004*N_REAL**4.00Q+0000-1.2277Q+0004*N_REAL**5.00Q+0000+7.15Q+000
&2*N_REAL**6.00Q+0000)-7.20Q+0001*I_REAL**7.00Q+0000*N_REAL*(2.4192
&Q+0005-6.17184Q+0005*N_REAL+6.1932Q+0005*N_REAL**2.00Q+0000-3.1517
&2Q+0005*N_REAL**3.00Q+0000+8.6458Q+0004*N_REAL**4.00Q+0000-1.2277Q
&+0004*N_REAL**5.00Q+0000+7.15Q+0002*N_REAL**6.00Q+0000)-1.20Q+0001
&*I_REAL**3.00Q+0000*N_REAL**5.00Q+0000*(1.8648Q+0005-5.04558Q+0005
&*N_REAL+5.34915Q+0005*N_REAL**2.00Q+0000-2.86714Q+0005*N_REAL**3.0
&0Q+0000+8.2786Q+0004*N_REAL**4.00Q+0000-1.2419Q+0004*N_REAL**5.00Q
&+0000+7.70Q+0002*N_REAL**6.00Q+0000)+8.40Q+0001*I_REAL**6.00Q+0000
&*N_REAL**2.00Q+0000*(3.3264Q+0005-8.51148Q+0005*N_REAL+8.56524Q+00
&05*N_REAL**2.00Q+0000-4.37088Q+0005*N_REAL**3.00Q+0000+1.20237Q+00
&05*N_REAL**4.00Q+0000-1.7126Q+0004*N_REAL**5.00Q+0000+1.001Q+0003*
&N_REAL**6.00Q+0000)+3.00Q+0001*I_REAL**4.00Q+0000*N_REAL**4.00Q+00
&00*(3.33648Q+0005-8.74602Q+0005*N_REAL+9.00363Q+0005*N_REAL**2.00Q
&+0000-4.69514Q+0005*N_REAL**3.00Q+0000+1.31969Q+0005*N_REAL**4.00Q
&+0000-1.9239Q+0004*N_REAL**5.00Q+0000+1.155Q+0003*N_REAL**6.00Q+00
&00))*DX**8.00Q+0000)/((-8.00Q+0000+N_REAL)*(-7.00Q+0000+N_REAL)*(-
&6.00Q+0000+N_REAL)*(-5.00Q+0000+N_REAL)*(-4.00Q+0000+N_REAL)*(-3.0
&0Q+0000+N_REAL)**2.00Q+0000*(-2.00Q+0000+N_REAL)**3.00Q+0000*(-1.0
&0Q+0000+N_REAL)**4.00Q+0000)
 MATRIX_QQ_DIAGONALS( 1,10,I) =(1.00Q-0001*(2.00Q+0000*I_REAL-1.00Q
&+0000*N_REAL)*((-2.00Q+0000+N_REAL)**2.00Q+0000*(-1.00Q+0000+N_REA
&L)**4.00Q+0000*N_REAL**8.00Q+0000*(1.20Q+0001-7.00Q+0000*N_REAL+N_
&REAL**2.00Q+0000)-4.00Q+0000*I_REAL*(-1.00Q+0000+N_REAL)**3.00Q+00
&00*N_REAL**7.00Q+0000*(-1.224Q+0004+1.699Q+0004*N_REAL-9.285Q+0003
&*N_REAL**2.00Q+0000+2.575Q+0003*N_REAL**3.00Q+0000-3.69Q+0002*N_RE
&AL**4.00Q+0000+2.20Q+0001*N_REAL**5.00Q+0000)+2.00Q+0000*I_REAL**2
&.00Q+0000*(-1.00Q+0000+N_REAL)**2.00Q+0000*N_REAL**6.00Q+0000*(1.2
&9456Q+0006-2.41466Q+0006*N_REAL+1.82147Q+0006*N_REAL**2.00Q+0000-7
&.14695Q+0005*N_REAL**3.00Q+0000+1.56029Q+0005*N_REAL**4.00Q+0000-1
&.8236Q+0004*N_REAL**5.00Q+0000+9.02Q+0002*N_REAL**6.00Q+0000)+1.00
&Q+0001*I_REAL**8.00Q+0000*(8.70912Q+0006-2.97181440Q+0007*N_REAL+4.
&17911040Q+0007*N_REAL**2.00Q+0000-3.1732968Q+0007*N_REAL**3.00Q+000
&0+1.43104Q+0007*N_REAL**4.00Q+0000-3.951350Q+0006*N_REAL**5.00Q+000
&0+6.57944Q+0005*N_REAL**6.00Q+0000-6.0977Q+0004*N_REAL**7.00Q+0000
&+2.431Q+0003*N_REAL**8.00Q+0000)-4.00Q+0001*I_REAL**7.00Q+0000*N_R
&EAL*(8.70912Q+0006-2.9718144Q+0007*N_REAL+4.1791104Q+0007*N_REAL**
&2.00Q+0000-3.1732968Q+0007*N_REAL**3.00Q+0000+1.43104Q+0007*N_REAL
&**4.00Q+0000-3.951350Q+0006*N_REAL**5.00Q+0000+6.57944Q+0005*N_REAL
&**6.00Q+0000-6.0977Q+0004*N_REAL**7.00Q+0000+2.431Q+0003*N_REAL**8
&.00Q+0000)-4.00Q+0000*I_REAL**3.00Q+0000*N_REAL**5.00Q+0000*(8.527
&68Q+0006-3.087108Q+0007*N_REAL+4.599924Q+0007*N_REAL**2.00Q+0000-3
&.6917995Q+0007*N_REAL**3.00Q+0000+1.7553691Q+0007*N_REAL**4.00Q+00
&00-5.102706Q+0006*N_REAL**5.00Q+0000+8.951570Q+0005*N_REAL**6.00Q+0
&000-8.7705Q+0004*N_REAL**7.00Q+0000+3.718Q+0003*N_REAL**8.00Q+0000
&)-2.00Q+0001*I_REAL**5.00Q+0000*N_REAL**3.00Q+0000*(2.17728Q+0007-
&7.502112Q+0007*N_REAL+1.065309120Q+0008*N_REAL**2.00Q+0000-8.166572
&4Q+0007*N_REAL**3.00Q+0000+3.7172408Q+0007*N_REAL**4.00Q+0000-1.03
&590970Q+0007*N_REAL**5.00Q+0000+1.741438Q+0006*N_REAL**6.00Q+0000-1
&.630750Q+0005*N_REAL**7.00Q+0000+6.578Q+0003*N_REAL**8.00Q+0000)+2.
&00Q+0001*I_REAL**6.00Q+0000*N_REAL**2.00Q+0000*(2.757888Q+0007-9.4
&3493760Q+0007*N_REAL+1.33022880Q+0008*N_REAL**2.00Q+0000-1.0126550Q+0
&008*N_REAL**3.00Q+0000+4.57817360Q+0007*N_REAL**4.00Q+0000-1.267284
&9Q+0007*N_REAL**5.00Q+0000+2.115682Q+0006*N_REAL**6.00Q+0000-1.966
&38Q+0005*N_REAL**7.00Q+0000+7.865Q+0003*N_REAL**8.00Q+0000)+2.00Q+
&0000*I_REAL**4.00Q+0000*N_REAL**4.00Q+0000*(8.836128Q+0007-3.09335
&4Q+0008*N_REAL+4.4619396Q+0008*N_REAL**2.00Q+0000-3.47247735Q+0008
&*N_REAL**3.00Q+0000+1.603690910Q+0008*N_REAL**4.00Q+0000-4.53294310Q
&+0007*N_REAL**5.00Q+0000+7.731127Q+0006*N_REAL**6.00Q+0000-7.35265
&Q+0005*N_REAL**7.00Q+0000+3.0173Q+0004*N_REAL**8.00Q+0000))*DX**9.
&00Q+0000)/((-9.00Q+0000+N_REAL)*(-8.00Q+0000+N_REAL)*(-7.00Q+0000+
&N_REAL)*(-6.00Q+0000+N_REAL)*(-5.00Q+0000+N_REAL)*(-4.00Q+0000+N_R
&EAL)**2.00Q+0000*(-3.00Q+0000+N_REAL)**2.00Q+0000*(-2.00Q+0000+N_R
&EAL)**3.00Q+0000*(-1.00Q+0000+N_REAL)**5.00Q+0000)
```

Reference(s):

(1) Middleton, T.F., Balla, R.J., Baurle, R.A., Humphreys, W.M. and Wilson, L.G.: "The NASA Langley Isolator Dynamics Research Lab," 31st Airbreathing Joint Meeting, JANNAF, La Jolla, California, 2009
(2) White, J.A., and Morrison, J.H.: "Pseudo-Temporal Multi-Grid Relaxation Scheme for Solving the Parabolized Navier-Stokes Equations," AIAA Paper 1999-3360, June 1999
(3) Menter, F.R.: "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," AIAA Journal Vol. 32, No. 8, 1994, pp. 1598-1605
(4) S. Bernstein: "Demonstration du Theoreme de Weierstrass Fondee sur le Calcul Des Probabilities," Comm. Soc. Math. Kharkov, 13 (1912) 1-2
(5) Bhrawy, A., Doha and Saker, M.: "On the Derivatives of Bernstein Polynomials: An Application for the Solution of High Even-Order Differential Equations," Applied Mathematics Letters, Vol. 2011, DOI: 10.1155/2011/829543
(6) Hoerl, A. and Kennard, R.: "Ridge Regression: Biased Estimation for Nonorthogonal Problems," Technometrics, 12, 55-67 (1970)
(7) Kaiser, Henry F.: "The JK Method: a Procedure for Finding the Eigenvectors and Eigenvalues of a Real Symmetric Matrix." Comput. J. 15 (1972): 271-273.
(8) Wolfram Research, Inc., Mathematica, Version 13.0, Champaign, Il. (2021)

Symbols and Notation:

$\circ$	= Hadamard matrix product	
$\frac{d^q}{dx^q}$	= qth-order ordinary differential x-operator	
$\frac{\partial^q}{\partial x^q}$	= q^{th}-order partial differential y-operator	
$\frac{\partial^q}{\partial y^q}$	= q^{th}-order partial differential y-operator	
Δx	= x-length scale discretizing parameter	
Δy	= y-length scale discretizing parameter	
$\Vert\ \Vert_2^2$	= Matrix norm squared (Euclidean distance metric)	
$[\ ]$	= Matrix	
$[\ ]^T$	= Matrix transpose operator	
$[\ ]^{-1}$	= Matrix inverse operator	
$F(x)$	= Function with x-dependency	
$F_{(i,n)}^{(q)}$	= Discretized q-order derivative of F(x)	
$\mathfrak{I}_B^{(q)}(x)$	= Continuous q-order derivative of the Bernstein Polynomial expansion of F(x)	
$\mathfrak{I}_{B_{(i,n)}}^{(q)}$	= Discretized q-order derivative of the Bernstein Polynomial expansion of F(x)	
ρ	= Density thermodynamic variable	
ρ_L	= Density field boundary condition values	(column vector)
ρ_R	= Density field interior domain values	(column vector)
$(\hat{\rho}_R)_{(\mu_1,\mu_2)}$	= Least Squares interior domain density constrained-solution set	(column vector)
μ_1	= Constraint parameter	
μ_2	= Constraint parameter (Ridge Regression)	
$[\Sigma^+]$	= Singular Value Decomposition matrix containing the singular values	
$[\Pi_{(i,n)}^{(N,\Delta x)}]$	= Derivative transform matrix operator	
$\Omega_{(i,n)}^{(k)}$	= Modified Bernstein Polynomial derivative operator element	

a	= Lower spatial bound	
b	= Upper spatial bound	
d_{min}	= Constraint vector	(column vector)
$B_i^n(x)$	= Bernstein Polynomial	
$[C]$	= Constraint matrix	
CFD	= Computational Fluid Dynamics	
$\left[D_{(x)}^{(B)}\right]$	= Discretized Bernstein polynomial derived x-derivative matrix operator	
$\left[D_{(y)}^{(B)}\right]$	= Discretized Bernstein polynomial derived y-derivative matrix operator	
FORTRAN	= FORmula TRANslation computer language	
[I]	= Identity matrix	
i	= Integer	
k	= Integer	
m	= Integer	
$[M_L]$	= Boundary condition density response matrix	
$[M_R]$	= Interior domain density response matrix	
n	= Integer	
N	= Integer cut-off parameter for derivative transform matrix	
NASA	= National Aeronautics and Space Administration	
Pa	= Pascals (pressure unit)	
q	= Integer	
Q	= Singular Value Decomposition unit-vector matrix	
rho	= Density	
s	= Integer	
st	= such that	
SVD	= Singular Value Decomposition	
u	= Streamwise-velocity component/variable	
U	= Streamwise-velocity component/variable	
$[U_{CFD}]$	= Streamwise-velocity field matrix obtained from CFD	
v	= Transverse-velocity component/variable	
V	= Transverse-velocity component/variable	
$[V_{CFD}]$	= Transverse-velocity field matrix obtained from CFD	
VULCAN	= NASA computational algorithm that solves the Reynolds-averaged fluid-conservation equations	
x	= Streamwise-spatial variable	
$[X_{(\rho_L,\rho_R)}]$	= Density field matrix	
y	= Transverse-spatial variable	