

A polar turbulence invariant map with applicability to realisable machine learning turbulence models

James G. Wnek¹ , Christopher Schrock², Eric M. Wolf³ and Mitch Wolff¹

¹Wright State University, 3640 Colonel Glenn Hwy, Dayton, OH 45435, USA

²Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, USA

³Ohio Aerospace Institute, 22800 Cedar Point Road, Cleveland, OH 44142, USA

Corresponding author: James G. Wnek, james.wnek@wright.edu

(Received 17 October 2024; revised 27 May 2025; accepted 15 July 2025)

Invariant maps are a useful tool for turbulence modelling, and the rapid growth of machine learning-based turbulence modelling research has led to renewed interest in them. They allow different turbulent states to be visualised in an interpretable manner and provide a mathematical framework to analyse or enforce realisability. Current invariant maps, however, are limited in machine learning models by the need for costly coordinate transformations and eigendecomposition at each point in the flow field. This paper introduces a new polar invariant map based on an angle that parametrises the relationship of the principal anisotropic stresses, and a scalar that describes the anisotropy magnitude relative to a maximum value. The polar invariant map reframes realisability in terms of a limiting anisotropy magnitude, allowing for new and simplified approaches to enforcing realisability that do not require coordinate transformations or explicit eigendecomposition. Potential applications to machine learning-based turbulence modelling include post-processing corrections for realisability, realisability-informed training, turbulence models with adaptive coefficients and general tensor basis models. The relationships to other invariant maps are illustrated through examples of plane channel flow and square duct flow. Sample calculations are provided for a comparison with a typical barycentric map-based method for enforcing realisability, showing an average 62 % reduction in calculation time using the equivalent polar formulation. The results provide a foundation for new approaches to enforcing realisability constraints in Reynolds-averaged turbulence modelling.

Key words: turbulence modelling, computational methods, machine learning

1. Introduction

The Reynolds stress is an important tensor in the Reynolds-averaged Navier–Stokes (RANS) equations, critical to the computational modelling of many practical aerodynamic flows. In recent years, machine learning (ML) has risen to prominence as a potential avenue to improve turbulence models using large amounts of experimental and computational data (Duraissamy, Iaccarino & Xiao 2019).

Although a great advantage of machine learning approaches is the lack of assumptions about the proper form of the model, it is well known that enforcing physical principles is an effective way to improve accuracy and generalisation capability when training data are limited. For example, Ling *et al.* (2016a) showed that embedding rotational invariance in the model architecture required a thousandth of the data needed to teach the model rotational invariance, by eliminating the need to augment the training data with rotations along each axis. Similarly, Franceschini, Sipp & Marquet (2020) found that on small training datasets, a turbulence model source term correction outperformed a momentum source term correction, but the latter surpassed the former with larger training datasets. They attributed this to the greater constraints on the turbulence model correction term, which limited the impact of poor predictions at the cost of a lower ceiling on the performance.

For turbulence modelling, realisability is an important guiding physical principle. In this context, realisability refers to a set of constraints on the Reynolds stress tensor that are necessary for a state of turbulence to be physically possible. Schumann (1977) showed that the Reynolds stress tensor must be positive semi-definite or, equivalently, that the eigenvalues are non-negative. The realisability constraints may be physically interpreted as requiring the non-negativity of energy.

The black-box nature of machine learning models makes realisability constraints relatively difficult to enforce compared with traditional modelling approaches with explicit formulations. Invariant maps have often been the preferred framework for enforcing realisability in ML turbulence modelling due to convenient mathematical properties and excellent interpretability (Tracey, Duraissamy & Alonso 2013). They are a way of representing tensors by creating a coordinate system of invariants, or quantities that do not change under rotations of the coordinate system (Irgens 2019). By using invariant maps, all states of turbulence can be mapped to a finite region, providing a unified framework for visualisation, analysis and enforcement of realisability constraints.

The earliest invariant map used in the field of turbulence was introduced by Lumley & Newman (1977) using the second and third invariants of the normalised Reynolds stress anisotropy tensor. It was later improved by Choi & Lumley (2001) by using the square and cube roots of the second and third invariants, so that a linear return to isotropy would form a straight line on the map. Both remain highly nonlinear, however, which can lead to misleading visualisations of the proximity to limiting states of turbulence.

Alternatively, linear invariant maps can be constructed from the eigenvalues. Lumley (1978) introduced the first of this type using the maximum and middle eigenvalues of the normalised anisotropy tensor. Banerjee *et al.* (2007) generalised the eigenvalue approach by constructing a barycentric coordinate system of the eigenvalues. The barycentric mapping improved on previous invariant maps in several ways. First, it equally weighted all the limiting states to provide an undistorted view of the anisotropy. Second, the linearity of the mapping allowed for direct, consistent interpolation between turbulent states (Banerjee *et al.* 2007). Finally, barycentric coordinates have the same form and properties as the red, green, blue (RGB) triplets used to define pixel colours, allowing the coordinates to be directly used as colour values in field visualisations of the anisotropy (Emory & Iaccarino 2014).

Invariant maps are frequently used in ML turbulence modelling applications due to their interpretability and mathematical properties. The barycentric map has been favoured due to the linearity and the simple form of the realisability constraints in barycentric coordinates. For example, Emory, Larsson & Iaccarino (2013) introduced perturbations in the barycentric mapping to investigate uncertainty in RANS models while maintaining realisability of the perturbed tensors. Tracey *et al.* (2013) developed a model that corrects discrepancies between RANS and DNS predictions of the anisotropy by shifting the turbulent state in barycentric coordinates. Wang, Wu & Xiao (2017) and Wu, Xiao & Paterson (2018) built on that approach by also predicting discrepancies in the eigenvectors and turbulent kinetic energy.

Although current invariant maps have been useful in the study of turbulence and the development of new ML-based models, the requirement of converting to and from invariant coordinates limits the range of models to which they can be applied. Furthermore, methods using barycentric coordinates require performing an eigendecomposition at every point in the flow field, which can be costly to compute. Alternative mappings like the ξ - η map are rarely, if ever, directly used in ML models due to their highly nonlinear nature (Banerjee *et al.* 2007). It would therefore be advantageous to ML turbulence modelling if an invariant map could be developed that bypasses the need for eigendecomposition without introducing strong nonlinearities.

The current work proposes a novel polar invariant map representation that represents the Reynolds stress anisotropy in terms of an angle and a fraction of the maximum realisable anisotropy magnitude. We will show that this new vector-like representation reframes realisability as a limit on the anisotropy magnitude, reframing the problem of imposing realisability constraints as a scaling problem. It will be demonstrated by comparison with direct numerical simulation (DNS) data that the new invariant map retains the interpretability of previous maps, as well as highlighting different aspects of the flow physics. We will show that the reframing of realisability as a limiting magnitude removes the need for eigendecomposition and transformation matrices, allowing for new and more efficient approaches to realisability in machine learning turbulence modelling.

The rest of this paper is divided into four sections. Section 2 discusses the derivation of the polar invariant map. Section 3 discusses the relationship between the new invariant map and previous invariant maps, both mathematically and by comparison with DNS data. Section 4 discusses the potential applications to ML turbulence modelling. Finally, § 5 discusses the implications and directions for future research.

2. Background

2.1. Reynolds stress and realisability

The Reynolds stress tensor can be decomposed into isotropic and anisotropic components according to

$$\tau_{ij} = \overline{u'_i u'_j} = a_{ij} + \frac{2}{3} k \delta_{ij}. \quad (2.1)$$

Here, $\tau_{ij} = \overline{u'_i u'_j}$ is the Reynolds stress tensor, where u'_i and u'_j represent the fluctuating velocity components. The tensor a_{ij} is the anisotropy tensor, k is the turbulent kinetic energy and $(2/3)k\delta_{ij}$ is the isotropic stress (Lumley 1978). The operator, δ_{ij} , is the Kronecker delta.

The anisotropy tensor is a second-order symmetric tensor with five independent components and zero trace. It can be normalised by $2k$ to obtain the normalised anisotropy tensor,

$$b_{ij} = \frac{a_{ij}}{2k} = \frac{\overline{u'_i u'_j}}{2k} - \frac{1}{3} \delta_{ij}. \quad (2.2)$$

The normalised anisotropy tensor, b_{ij} , is constrained by the requirement that the Reynolds stress tensor is positive semi-definite or, equivalently, that the eigenvalues are non-negative (Schumann 1977). The realisability constraints may be physically interpreted as a condition that the turbulent kinetic energy be non-negative.

The non-negativity of the eigenvalues implies that the minimum eigenvalue of the Reynolds stress tensor is 0. The maximum occurs when only one eigenvalue is non-zero, as that eigenvalue will be equal to $2k$ by definition of the turbulent kinetic energy. Substituting the maximum and minimum eigenvalues into (2.2) allows realisability to be written in terms of the eigenvalues of the normalised anisotropy tensor (Lumley 1978),

$$-\frac{1}{3} \leq \lambda_3 \leq \lambda_2 \leq \lambda_1 \leq \frac{2}{3}. \quad (2.3)$$

Here, $\lambda_3 \leq \lambda_2 \leq \lambda_1$ are the eigenvalues of b_{ij} . Note that it is unnecessary to consider λ_2 when determining realisability because it will always be between λ_1 and λ_3 (Banerjee *et al.* 2007). Additionally, as a consequence of b_{ij} being traceless, the eigenvalues cannot all have the same sign. From this, it is clear that $\lambda_1 \geq 0$ and $\lambda_3 \leq 0$.

Because of the bounds on b_{ij} , it is possible to represent all turbulent states in a finite region using a coordinate system constructed from the invariants of the normalised anisotropy tensor.

2.2. Current invariant maps

Invariants are quantities associated with a tensor that do not change when the coordinate system is rotated. The principal invariants for an arbitrary second-order symmetric tensor, σ_{ij} , are

$$I_1 = \sigma_{ii}, \quad I_2 = \frac{1}{2} \left(\sigma_{ii}^2 - \sigma_{ij} \sigma_{ij} \right), \quad I_3 = \det(\sigma_{ij}). \quad (2.4)$$

These three invariants are referred to as the first, second and third invariants (Irgens 2019). For the anisotropic part of a second-order symmetric tensor, the first invariant is zero and the other two principal invariants become (Chakrabarty 2006)

$$J_2 = \frac{1}{2} \sigma_{ij}^{dev} \sigma_{ij}^{dev} = \frac{1}{3} I_1^2 - I_2, \quad J_3 = \det(\sigma_{ij}^{dev}) = \frac{2I_1^3 - 9I_1 I_2 + 27I_3}{27}. \quad (2.5)$$

Here, J_2 and J_3 are the principal invariants of σ_{ij}^{dev} , which is the deviatoric part of σ_{ij} .

Three invariant maps have been commonly used in the study of turbulence and turbulence modelling. The first was introduced by Lumley & Newman (1977) using the second and third invariants of the anisotropy tensor,

$$J_{2,b} = \frac{1}{2} b_{ij} b_{ij}, \quad J_{3,b} = \det(b_{ij}). \quad (2.6)$$

The subscript b indicates that these invariants are associated with b_{ij} . Lumley used these invariants to form a region with two curved sides and one straight side that contained all realisable states of turbulence.

The second invariant map was introduced by Choi & Lumley (2001) by defining two new invariants derived from the principal invariants,

$$\xi^3 = J_{3,b}/2, \quad \eta^2 = J_{2,b}/3. \quad (2.7)$$

The invariants, ξ and η , form a mapping with two straight sides and one curved side. This modification was motivated by the study of the return to isotropy of homogeneous turbulence. By using ξ and η , a linear return to isotropy as proposed by Rotta (1951) would show as a straight line through the origin, allowing any nonlinearities to be clearly visualised. Both invariant maps have been used in the development of return-to-isotropy models for Reynolds Stress Transport Models (Lumley & Newman 1977; Choi & Lumley 2001).

While these mappings were highly nonlinear, which can result in misleading visualisations of the turbulent state (Banerjee *et al.* 2007). For example, in an early DNS study of fully developed turbulent channel flow, the nonlinear mapping made it appear that the turbulence was nearly isotropic at the centreline, when in fact, it was far from isotropic (Antonia, Kim & Browne 1991). A similar conclusion was made in a different study of turbulent pipe flow (Krogstad & Torbergsen 2000).

The most recent invariant map was proposed by Banerjee *et al.* (2007) to provide unbiased visualisations of the turbulent state and as a formulation for exact interpolation between states. Banerjee *et al.* used the eigenvalues to construct a barycentric coordinate system where the turbulent states were convex combinations of the limiting states of turbulence, defined by

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} x_{1c} & x_{2c} & x_{3c} \\ y_{1c} & y_{2c} & y_{3c} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}. \quad (2.8)$$

Here, x_{1c} , y_{1c} and so on are the Cartesian coordinates of the corners of the map. Additionally, C_1 , C_2 and C_3 are the barycentric coordinates for the first, second and third corners, respectively. The coordinates x_B and y_B are the Cartesian coordinates of the barycentric map. The barycentric coordinates are defined by

$$C_1 = \lambda_1 - \lambda_2, \quad C_2 = 2(\lambda_2 - \lambda_3), \quad C_3 = 1 + 3\lambda_3, \quad (2.9)$$

where $C_1 + C_2 + C_3 = 1$, $C_1, C_2, C_3 \geq 0$.

All of these invariant maps are depicted in figure 1. In any invariant map, the corners represent the three limiting states of turbulence: one-component (1C), two-component axisymmetric (2C) and isotropic (3C). The 1C–2C line indicates states of two-component turbulence, but not necessarily axisymmetric. The 1C–3C line indicates axisymmetric expansion (AE), and the 2C–3C line indicates axisymmetric contraction (AC) (Banerjee *et al.* 2007). It is important to note that early researchers have used these terms to refer to both the magnitude of the principal stresses and the physical shape of the eddies. This was a source of confusion as the identification of the sides corresponding to axisymmetric expansion and contraction would flip depending on which interpretation was used (Simonsen & Krogstad 2005).

Here, we use the convention used by Lumley & Newman (1977), Choi & Lumley (2001) and Banerjee *et al.* (2007), which refer to the relative magnitudes of principal Reynolds stresses. One-component means one principal stress is non-zero; two-component, axisymmetric means two principal stresses are non-zero and equal in magnitude; and isotropic means all principal stresses are non-zero and equal. The terms axisymmetric expansion and contraction in this context refer to the behaviour of the principal Reynolds

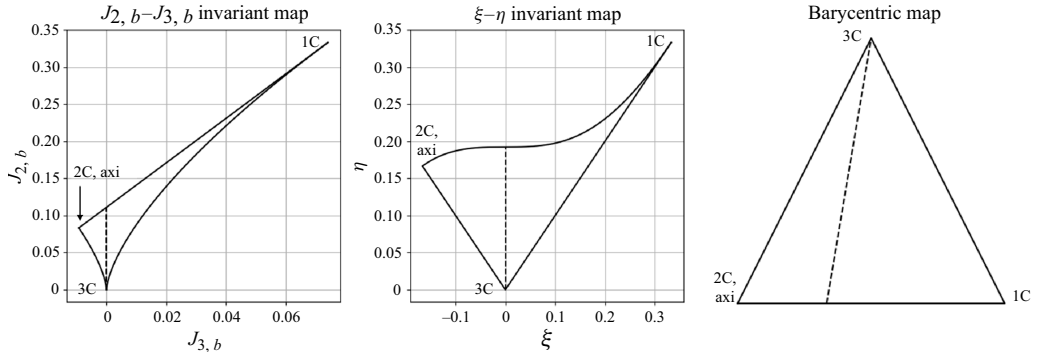


Figure 1. A comparison of the $J_{2,b}$ - $J_{3,b}$, ξ - η and barycentric maps. 1C indicates one-component turbulence, 2C-axi indicates two-component axisymmetric turbulence, and 3C indicates isotropic turbulence. The dashed line indicates the plane-strain limit.

stresses as the anisotropy magnitude shrinks. If the two equal principal stresses grow, it is in a state of expansion, and if they shrink, it is in a state of contraction.

3. Polar invariant map construction

3.1. Analytical form of eigenvalues

It is informative to begin from the eigenvalues of the normalised anisotropy tensor, b_{ij} . The eigenvalues of the arbitrary second-order, symmetric tensor, σ_{ij} , are the solutions to the cubic equation (Pope 2000),

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0. \quad (3.1)$$

Here, I_1 , I_2 and I_3 are the principal invariants of σ_{ij} (Irgens 2019). The solution to this equation can be obtained through Vieta's trigonometric formula for the roots of a cubic polynomial. This gives the following solution for the eigenvalues in terms of the invariants (Zucker 2008):

$$\lambda_n = \frac{I_1}{3} + \sqrt{\frac{4J_2}{3}} \cos \left[\theta - \frac{2\pi(n-1)}{3} \right], \quad n = 1, 2, 3, \quad (3.2a)$$

$$\theta = \frac{1}{3} \arccos \left(\frac{3\sqrt{3}}{2} \frac{J_3}{J_2^{3/2}} \right). \quad (3.2b)$$

In (3.2), θ is an invariant that exists for all non-zero, symmetric, 3×3 tensors (Chakrabarty 2006). Although we are unaware of any examples of its usage in turbulence modelling, it is an established invariant in solid mechanics that is used in the Haigh–Westergaard coordinate system for modelling material behaviour and visualising stress states (Kassir 2017).

Substituting in the definitions of I_1 , J_2 and J_3 , (3.2a) may be rewritten in terms of σ_{ij} and θ as

$$\lambda_n = \frac{\sigma_{ii}}{3} + \sqrt{\frac{2}{3} \sigma_{ij}^{dev} \sigma_{ij}^{dev}} \cos \left[\theta - \frac{2\pi(n-1)}{3} \right], \quad n = 1, 2, 3. \quad (3.3)$$

An additional simplification can be made by recognising $\sigma_{ij}^{dev} \sigma_{ij}^{dev}$ as the square of Frobenius norm, $\|\sigma_{ij}^{dev}\|^2$. The general solution simplifies to

$$\lambda_n = \frac{\sigma_{ii}}{3} + \frac{\sqrt{6}}{3} \|\sigma_{ij}^{dev}\| \cos \left[\theta - \frac{2\pi(n-1)}{3} \right], \quad n = 1, 2, 3. \quad (3.4)$$

Using the same simplification for (3.2b) yields

$$\theta = \frac{1}{3} \arccos \left(\frac{3\sqrt{3}}{2(1/2)^{3/2}} \frac{\det(\sigma_{ij}^{dev})}{\|\sigma_{ij}^{dev}\|^3} \right) = \frac{1}{3} \arccos \left(3\sqrt{6} \det \left(\frac{\sigma_{ij}^{dev}}{\|\sigma_{ij}^{dev}\|} \right) \right). \quad (3.5)$$

In (3.5), bringing the norm of σ_{ij}^{dev} into the determinant follows from the property that $\det(cA) = c^n \det(A)$ for any scalar c and matrix A of rank n (Baker & Porteous 1990).

Equations (3.4) and (3.5) can now be applied to the normalised anisotropy tensor by substituting b_{ij} for σ_{ij} . Because b_{ij} has zero trace, the first term on the right-hand side of (3.4) is eliminated and $b_{ij} = \sigma_{ij}^{dev}$. Making these substitutions gives

$$\lambda_n = \frac{\sqrt{6}}{3} \|b_{ij}\| \cos \left[\theta - \frac{2\pi(n-1)}{3} \right], \quad n = 1, 2, 3. \quad (3.6a)$$

$$\theta = \frac{1}{3} \arccos \left(3\sqrt{6} \det \left(\frac{b_{ij}}{\|b_{ij}\|} \right) \right) \in \left[0, \frac{\pi}{3} \right]. \quad (3.6b)$$

The bounds on θ are a result of the limited range of the arccosine function. Neglecting cases in which $\|b_{ij}\| = 0$, the restrictions on the domain of arccosine are satisfied by all symmetric, deviatoric tensors (Chakrabarty 2006). A proof of this may be found in Appendix A.

Evaluating (3.6a) for λ_1 , λ_2 and λ_3 finally gives explicit forms for the eigenvalues of b_{ij} ,

$$\lambda_1 = \frac{\sqrt{6}}{3} \|b_{ij}\| \cos \theta, \quad (3.7a)$$

$$\lambda_2 = \frac{\sqrt{6}}{3} \|b_{ij}\| \cos \left[\theta - \frac{2\pi}{3} \right], \quad (3.7b)$$

$$\lambda_3 = \frac{\sqrt{6}}{3} \|b_{ij}\| \cos \left[\theta + \frac{2\pi}{3} \right], \quad (3.7c)$$

It should be noted that θ is undefined when $\|b_{ij}\| = 0$ due to $b_{ij}/\|b_{ij}\|$ becoming indeterminate. The coefficient $\|b_{ij}\|$ in (3.7a)–(3.7c) drives the eigenvalues to zero, so the value of θ does not affect the output. This occurs for purely isotropic flow, where the normalised anisotropy tensor becomes a zero matrix. Such a case can be handled numerically by explicitly setting the eigenvalues to zero when $\|b_{ij}\| = 0$.

3.2. Normalisation of anisotropy magnitude

In (3.7), $\|b_{ij}\|$ and θ form a set of polar coordinates where $\|b_{ij}\|$ describes the anisotropy magnitude and θ describes the relationship of the eigenvalues or principal stresses. This is most easily interpreted by viewing the eigenvalues as forming a vector in the principal axes, where θ parametrises the direction of the vector. Figure 2 depicts this scenario with the unit vector, $\hat{\lambda}$. In figure 2, the axes are the eigenvectors e_1 , e_2 and e_3 , associated with the principle stresses, λ_1 , λ_2 and λ_3 . The circle represents the plane where $\lambda_1 + \lambda_2 + \lambda_3 = 0$, and its edge is defined by the unit vectors of the principal anisotropic stresses, denoted as $\hat{\lambda}$. Equation (3.7a) constrains the possible combinations of the principal stresses to the sector bounded by the blue lines.

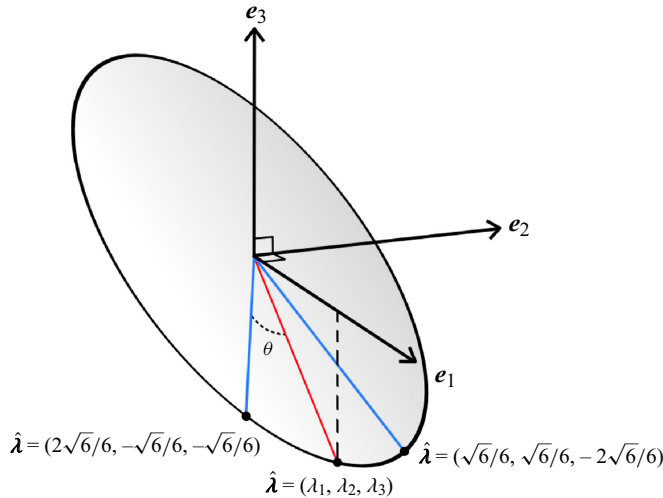


Figure 2. A geometric interpretation of θ . The axes are the eigenvectors of b_{ij} , the blue lines indicate the bounds on θ and the red line is a unit vector.

Converting from standard Cartesian invariant maps to a polar invariant map provides an alternative perspective on the realisability bounds as a limit on the normalised anisotropy magnitude. Using this concept of realisability as a limit on the normalised anisotropy magnitude, we can normalise $||b_{ij}||$ according to its maximum realisable magnitude at the same θ as

$$\alpha \equiv \frac{||b_{ij}||}{||b_{ij}||_{\max}}. \quad (3.8)$$

Here, α is the fraction of the maximum anisotropy. It can be interpreted as a normalised distance from isotropy in that a value of zero implies perfectly isotropic turbulence, while a value of one implies maximally anisotropic turbulence. In this context, θ takes on the role of determining the limiting anisotropy magnitude.

The maximum value of $||b_{ij}||$ can be determined as a function of θ by substituting (3.7a) and (3.7c) into the inequality in (2.3). This gives the set of inequalities,

$$\lambda_1 = \frac{\sqrt{6}}{3} ||b_{ij}|| \cos \theta \leq 2/3, \quad (3.9a)$$

$$\lambda_3 = \frac{\sqrt{6}}{3} ||b_{ij}|| \cos \left[\theta + \frac{2\pi}{3} \right] \geq -1/3. \quad (3.9b)$$

A negative can be factored out of λ_3 by introducing a phase shift of π radians. Then, the constraints on $||b_{ij}||$ may be expressed as

$$||b_{ij}|| \leq \frac{\sqrt{6}}{3 \cos \theta}, \quad (3.10a)$$

$$||b_{ij}|| \leq \frac{\sqrt{6}}{6 \cos \left[\theta - \frac{\pi}{3} \right]}. \quad (3.10b)$$

It is simple to verify that the first expression always exceeds or equals the second for $\theta \in [0, \pi/3]$. Thus, the limiting anisotropy magnitude is determined by the more restrictive

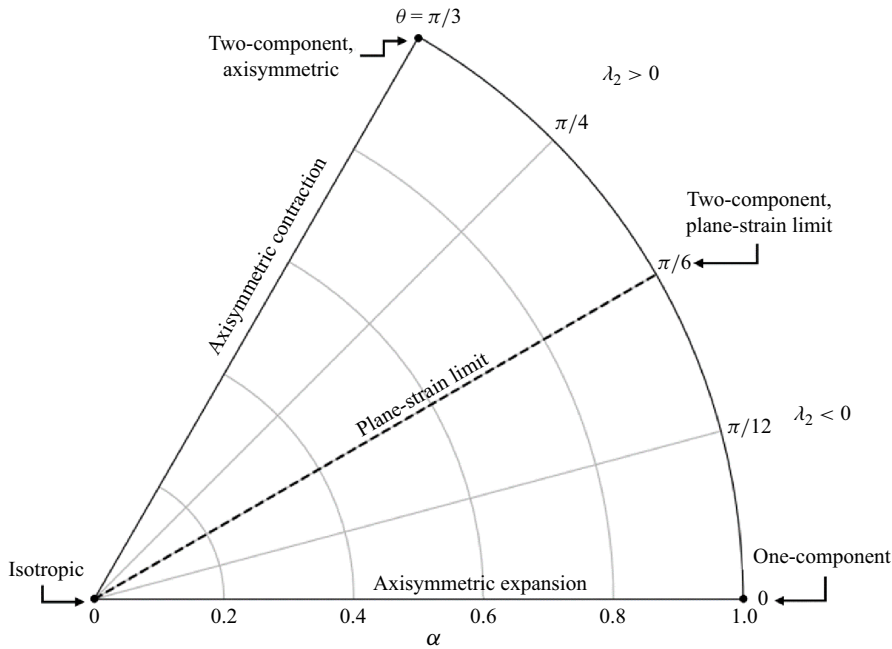


Figure 3. An annotated diagram of the polar α - θ invariant map.

inequality, (3.10b), to be

$$||b_{ij}||_{\max} = \frac{\sqrt{6}}{6 \cos \left[\theta - \frac{\pi}{3} \right]}. \quad (3.11)$$

Using $||b_{ij}||_{\max}$, an explicit form for α may be obtained as

$$\alpha = \frac{||b_{ij}||}{||b_{ij}||_{\max}} = \sqrt{6} ||b_{ij}|| \cos \left(\theta - \frac{\pi}{3} \right). \quad (3.12)$$

3.3. Polar invariant map properties

With α and θ known, the polar invariant map can be plotted as

$$x_p = \alpha \cos \theta, \quad y_p = \alpha \sin \theta, \quad (3.13)$$

where x_p and y_p are the Cartesian x and y coordinates, respectively. An annotated diagram of the polar invariant map is depicted in figure 3. Like all invariant maps, the three corners correspond to the limiting states of turbulence: one-component; two-component, axisymmetric; and isotropic. The line $\theta = 0$ represents states of axisymmetric expansion and the line $\theta = \pi/3$ represents states of axisymmetric contraction. Here, the terms axisymmetric expansion and contraction refer to the behaviour of the principal Reynolds stresses as the anisotropy magnitude shrinks. If the two equal principal stresses grow, it is in a state of expansion, and if they shrink, it is in a state of contraction (Banerjee *et al.* 2007).

Splitting the invariant map into two halves is the plane-strain line at $\theta = \pi/6$. Below the plane-strain line, λ_2 is less than 0, implying λ_1 has a greater magnitude than λ_3 . Conversely, above the plane-strain line, λ_2 is greater than 0, implying λ_1 has a smaller magnitude than λ_3 . At the plane-strain line, $\lambda_2 = 0$, so λ_1 and λ_3 have equal magnitude.

Lines of constant θ on the map imply a fixed ratio of the principal stresses, while lines of constant α imply a fixed ratio of the actual and maximum anisotropy magnitudes. Additional insight can be gained by substituting the definition of α into (3.7) to obtain eigenvalue formulae in terms of α and θ ,

$$\lambda_1 = \frac{\alpha}{3} \cos \theta \sec \left[\theta - \frac{\pi}{3} \right], \quad (3.14a)$$

$$\lambda_2 = \frac{\alpha}{3} \cos \left[\theta - \frac{2\pi}{3} \right] \sec \left[\theta - \frac{\pi}{3} \right], \quad (3.14b)$$

$$\lambda_3 = -\frac{\alpha}{3}. \quad (3.14c)$$

Two important observations may be made from (3.14). First, λ_3 depends solely on α . Additionally, when $\alpha = 1$, λ_3 reaches its minimum value of $-1/3$. Thus, the maximum anisotropy magnitude always occurs when $\lambda_3 = -1/3$. This corresponds to the two-component turbulence arc on the polar invariant map.

Second, because α is positive by definition and θ is defined on $[0, \pi/3]$ for all symmetric, deviatoric tensors, all non-realisable states must have $\alpha > 1$. Consequently, knowing α , or equivalently λ_3 , is sufficient to determine whether a point in the flow field violates realisability. Furthermore, all combinations of $\alpha \in [0, 1]$ and $\theta \in [0, \pi/3]$ are realisable states.

The preceding observations lead to two convenient properties. First, the nearest realisable point to any non-realisable point is one with the same θ , but at $\alpha = 1$. Second, because of the form of the eigenvalues, any non-realisable state, $\alpha > 1$, can be brought to the nearest realisable state, $\alpha = 1$, by simply scaling the original tensor by the multiplicative inverse, $1/\alpha$. More generally, a tensor can be shifted in the polar invariant map along a line of constant θ using a scaling factor of

$$\beta = \frac{\alpha_1}{\alpha_0}, \quad (3.15)$$

where α_0 is the original α value and α_1 is the target α . This provides a direct link to turbulence models that can be written as a nonlinear eddy viscosity model by representing realisability limits in terms of a scaling factor analogous to the eddy viscosity. Such an approach would be especially useful for the myriad of machine learning-based turbulence modelling concepts, which are generally challenging to enforce realizability with. The potential applications will be elaborated upon in § 4.

4. Comparison with other invariant maps

An initial comparison between the polar invariant map and other invariant maps can be made by using the barycentric coordinates as pixel values for each point in the maps. This is accomplished by converting each invariant map's coordinates to the equivalent barycentric coordinates, then forming a list of triplets of the barycentric coordinates at each point of an invariant map. The list of triplets may then be used as a list of red, green and blue colour values (RGB triplets) to directly define the colours of the pixels in the plot (Emory & Iaccarino 2014). The polar invariant map will be compared with the Lumley triangle and barycentric map using red for one-component turbulence, green for axisymmetric two-component turbulence and blue for isotropic turbulence. The results are displayed in figure 4. Algorithm 1 in Appendix B gives pseudocode for how the list of RGB triplets is generated.

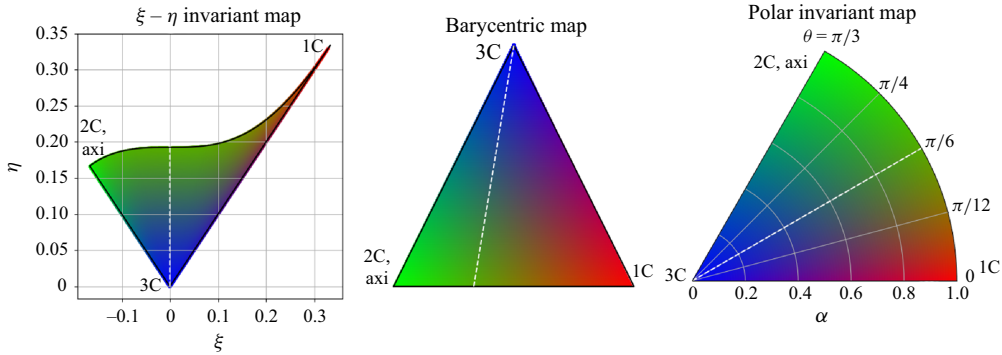


Figure 4. A comparison of the ξ - η , barycentric and polar invariant maps using barycentric coordinates as pixel values. The plane-strain limit is overlaid as a dashed line.

Figure 4 shows that two-component turbulence takes up a larger region of the polar invariant map compared with the barycentric map, causing the one-component corner to become compressed. In comparison, there is little effect on the isotropic turbulence corner. Thus, in terms of turbulence as a combination of the limiting states, the polar map tends to give greater weight to two-component turbulence. Despite the unequal weighting, it remains far less distorted than the one-component corner on the ξ - η invariant map.

4.1. Mathematical relationship

To more precisely show the relationship between the polar invariant map and the other invariant maps, the equations for their coordinates can be rewritten in terms of α and θ . As a first step, (2.6) may be rewritten in terms of λ_1 and λ_3 as

$$J_{2,b} = \lambda_1^2 + \lambda_1 \lambda_3 + \lambda_3^2, \quad J_{3,b} = -\lambda_1^2 \lambda_3 - \lambda_1 \lambda_3^2. \quad (4.1)$$

Next, define $\zeta = \cos(\theta) \sec(\theta - \pi/3)$. Substituting (3.14a) and (3.14c) into (4.1) allows the invariants to be written in terms of α and θ ,

$$J_{2,b} = \left(\frac{\alpha}{3}\right)^2 [\zeta^2 - \zeta + 1], \quad J_{3,b} = \left(\frac{\alpha}{3}\right)^3 [\zeta^2 - \zeta]. \quad (4.2)$$

Note that ζ is equivalent to λ_1/λ_3 . Using the invariants as given in (4.2), ξ and η become

$$\eta = \frac{\alpha}{3} \left[\frac{\zeta^2 - \zeta + 1}{3} \right]^{1/2}, \quad \xi = \frac{\alpha}{3} \left[\frac{\zeta^2 - \zeta}{2} \right]^{1/3}. \quad (4.3)$$

It is clear from (4.3) that the nonlinearity in the ξ - η invariant map comes exclusively from the θ dependence. This indicates that the ξ - η invariant map represents the relationship between eigenvalues in a nonlinear manner, and the relationship between the actual and maximum anisotropy magnitude in a linear manner.

To develop mathematical relationships between the polar and barycentric maps, it is helpful to begin by using the property of the normalised anisotropy tensor having zero trace to eliminate λ_2 from the barycentric coordinates. This gives

$$C_1 = 2\lambda_1 + \lambda_3, \quad C_2 = -2(\lambda_1 + 2\lambda_3), \quad C_3 = 3\lambda_3 + 1. \quad (4.4)$$

Equations (3.14a) and (3.14c) can be substituted into (4.4) to obtain the barycentric coordinates in terms of α and θ ,

$$C_1 = \frac{\alpha}{3}(2\zeta - 1), \quad C_2 = \frac{2\alpha}{3}(2 - \zeta), \quad C_3 = 1 - \alpha. \quad (4.5)$$

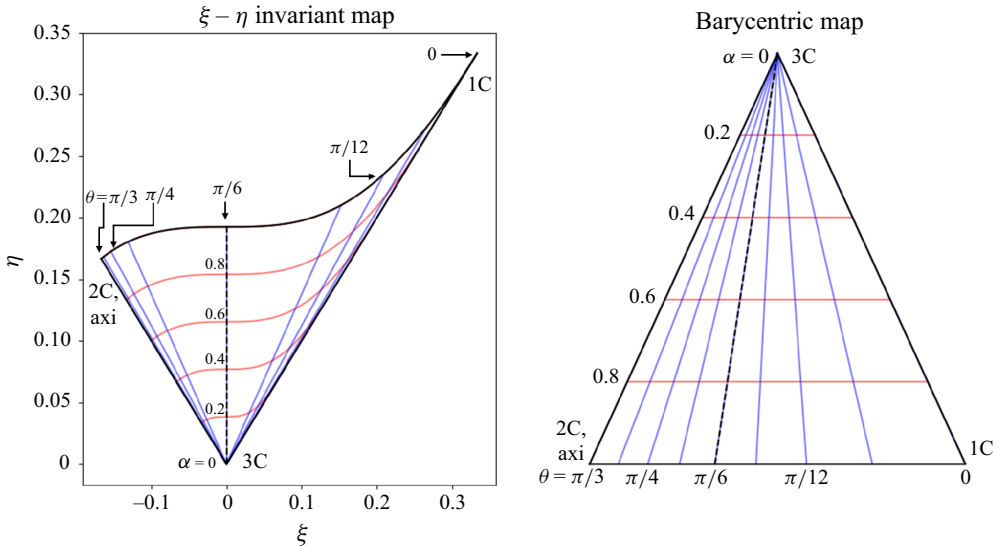


Figure 5. The ξ – η map and barycentric map overlaid with lines of α and θ . Red lines indicate constant α and blue lines indicate constant θ . The black dashed line indicates the plane-strain limit.

Note that C_3 is a linear function of only α . This is unsurprising as the C_3 axis in the barycentric map describes the level of isotropy. In contrast, C_1 and C_2 depend on both α and θ . This also makes sense as the C_1 and C_2 axes have both vertical and horizontal components.

Banerjee *et al.* (2007) defined the turbulence triangle's Cartesian coordinates using $(x_{1c}, y_{1c}) = (1, 0)$, $(x_{2c}, y_{2c}) = (0, 0)$ and $(x_{3c}, y_{3c}) = (1/2, \sqrt{3}/2)$. Using these coordinates for the corners of the barycentric map in (2.8) simplifies the transformation of barycentric to Cartesian coordinates to

$$x_B = C_1 + \frac{1}{2}C_3, \quad y_B = \frac{\sqrt{3}}{2}C_3. \quad (4.6)$$

Substituting (4.5) into (4.6) gives the Cartesian coordinates in terms of α and θ ,

$$x_B = \frac{1}{2} + \frac{\alpha}{3} \left(2\zeta - \frac{5}{2} \right), \quad y_B = \frac{\sqrt{3}}{2}(1 - \alpha). \quad (4.7)$$

In (4.7), similar to the barycentric coordinates, the Cartesian coordinates show that the vertical axis is solely dependent on α , while the horizontal axes are dependent on both. The dependence on α and θ is necessary because the range of x_B contracts as y_B increases.

Using (4.3) and (4.7), θ and α isolines can be plotted on the barycentric and ξ – η invariant maps to visualise the distortion relative to the polar map. This is depicted in figure 5.

The most significant difference between both the ξ – η and barycentric maps compared with the polar invariant map is the spacing of the θ isolines. In the barycentric map, the θ isolines tend to compress when moving from the one-component to two-component corners. This is consistent with the bias towards the two-component corner observed for the polar invariant map in figure 4. In comparison, the ξ – η map shows almost the opposite distortion in the θ isolines as the barycentric map. Here, the isolines are strongly pushed apart near the plane-strain limit at $\theta = \pi/6$ and are strongly compressed near the straight edges.

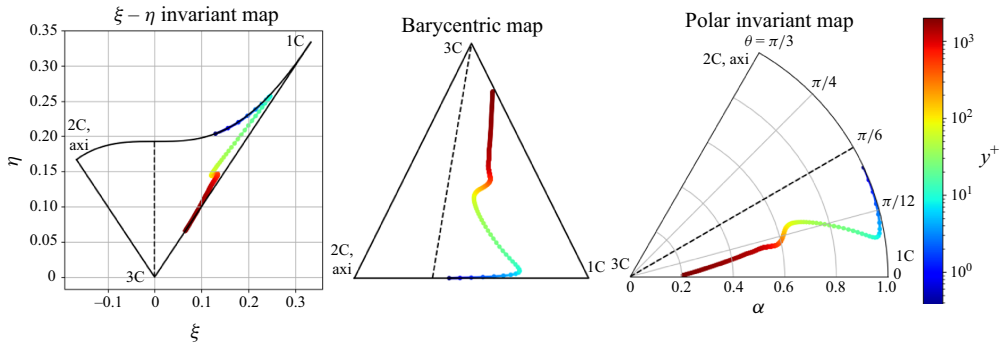


Figure 6. A comparison of fully developed channel flow at $Re_\tau = 2003$ in each invariant map. The data are coloured by y^+ . Lines interpolated from the DNS data are underlaid for clarity.

Interestingly, despite the differences in the shape of the α isolines, both maps preserve the linearity of α as the normalised distance from the isotropic corner. In the barycentric map, this manifests as α consisting of equally spaced straight lines. The ξ – η map instead displays α isolines that are equally spaced along any given θ isoline, but with a different spacing for each isoline.

Additionally, the polar invariant map sheds light on a disagreement between Mandler & Weigand (2022) and Blauw (2019) over whether all non-realisable points fall below the 1C–2C line on the barycentric map. The polar invariant map clarifies that any point with a valid θ can only be non-realisable by exceeding $\alpha = 1$, corresponding to points that fall below the 1C–2C line on the barycentric map. Points that fall outside the 1C–3C and 2C–3C lines correspond to points outside of $\theta = [0, \pi/3]$. This would require a violation of the range of (3.6b), which is valid for all symmetric, deviatoric tensors. It is clear then that the points Mandler & Weigand (2022) noted to fall outside the 1C–3C and 2C–3C lines during the solution convergence process must occur due to the violation of either the zero-trace or symmetric properties of the anisotropy tensor. This is unsurprising for a solution in the process of converging, as intermediate iterations frequently show violations of continuity, which result in a non-deviatoric anisotropy tensor.

4.2. Comparison with sample data

To illustrate the differences in the invariant maps on real data, we will compare DNS data of a fully developed channel flow and a cross-section of a square duct in each mapping.

4.2.1. Fully developed channel flow

First, consider a fully developed plane channel flow. A time-averaged DNS dataset was obtained from Hoyas & Jiménez (2006). The flow field is statistically two-dimensional, and the velocity and Reynolds stress profiles are self-similar (Pope 2000). Figure 6 depicts the DNS Reynolds stress profile for $Re_\tau = 2003$ visualised in each invariant map.

All three invariant maps show that the turbulence starts near the wall close to the plane-strain limit on the 1C–2C line, migrates towards the one-component corner and then moves closer to the centreline, before following a straight path to terminate near the 1C–3C side. Both the polar and barycentric maps show that the turbulence consistently moves closer to isotropy, though it remains significantly anisotropic throughout the entire profile. In contrast, the ξ – η map obscures this trend. Due to the expansion of the near plane-strain region, the profile appears significantly closer to the axisymmetric expansion line than it truly is. Similarly, the trajectory in the near-wall region appears to start much closer to the

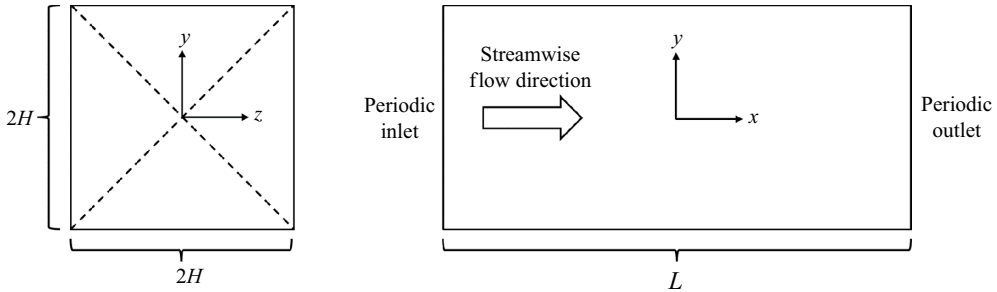


Figure 7. A diagram of the square duct flow. The flow field is symmetric across the dashed diagonals.

one-component corner in the ξ - η map than in any other mapping. It also shows a kink in the profile at approximately $y^+ = 250$ – 750 , which causes it to appear to be moving away from the isotropic corner, even though the barycentric and polar maps clearly show the profile continually moves towards that corner. These observations demonstrate how strongly nonlinear mappings like the ξ - η map can produce misleading visualisations and that the polar invariant map avoids this disadvantage.

While all of the maps provide information about the componentiality, the polar invariant map complements this interpretation by also making clear the relationship between eigenvalues and their relative magnitudes. The polar invariant map shows that the observation that the anisotropy approaches the isotropic corner is mathematically equivalent to stating the anisotropy magnitude is decreasing relative to its maximum value. Furthermore, the radial axis quantifies this change, indicating that the anisotropy magnitude at the centre of the channel is approximately 20 % of its maximum value for the given θ . It can also be seen that the relationship of the eigenvalues changes much more slowly past approximately $y^+ = 250$ based on the low variation in θ . Even the barycentric map, which provides very similar visualisations to the polar invariant map, appears to show a much greater rate of change in the relationship of the eigenvalues due to the expanded spacing of θ noted in figure 5. Thus, although the information is contained in other invariant maps, it is not readily accessible or quantifiable like in the polar invariant map.

4.2.2. Square duct flow

Next, we will compare the invariant maps for fully developed square duct flow. The DNS dataset was obtained from the turbulence modelling database of McConkey, Yee & Lien (2021), using the DNS dataset generated by Pinelli *et al.* (2010). A diagram of the square duct is displayed in figure 7. The flow domain has a height and width of $2H$, and a length of L . All of the square duct data are provided as time-averaged quantities.

Square ducts are one of the simplest geometries that exhibit secondary flows. Here, the secondary flows are pairs of counter-rotating vortices located in each corner of the duct. The DNS data from Pinelli *et al.* (2010) were collected for sixteen different Reynolds numbers; however, only the maximum Reynolds number, $Re = 3500$, will be shown here.

Figure 8 compares the Reynolds stress in a cross-stream plane in the ξ - η , barycentric and polar invariant maps. The points are coloured by the distance to the nearest wall, d/H . Note that $d/H = 1$ corresponds to the centreline of the duct. Similar to the channel flow case, each invariant map shows that the turbulence starts out on the 1C–2C line near the wall, then moves towards the centre before terminating near the isotropic-one-component line. The movement towards the plane-strain limit, as well as the distance between the turbulent trajectories and the axisymmetric expansion line, are clearly shown

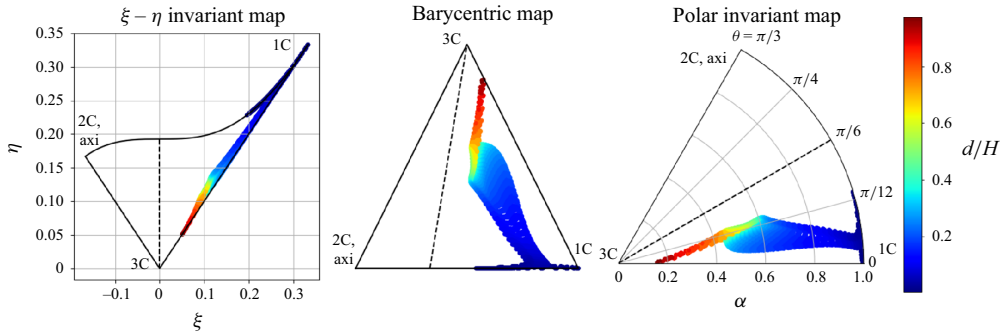


Figure 8. A comparison of the square duct Reynolds stress visualised in each invariant map. The points are coloured by the wall distance, d/H .

in the barycentric and polar maps but are nearly imperceptible in the ξ - η mapping. This again shows the limitations of the ξ - η invariant map for visualisation purposes.

In contrast to the channel flow case, all of the turbulent trajectories start considerably closer to the one-component corner. This is highlighted in the polar invariant map where the channel flow turbulent trajectory initiates near the wall at approximately $\theta = \pi/7$, whereas most points in the square duct never exceed $\theta = \pi/12$. Additionally, the trajectories fan out in the range of approximately 0.1 to $0.4d/H$, likely due to the profiles passing through different areas of the counter-rotating vortices.

To better visualise how [figure 8](#) relates to physical space, contour maps of the invariant coordinates can be plotted. This is displayed in [figure 9](#), overlaid with streamlines. Here, C_1 and C_3 from the barycentric map were plotted separately to make a fair comparison.

As expected from (4.5), the α and C_3 contours are mirror images of each other, the only difference being the reversal of the colour map. The C_1 , η and ξ contours also show several similarities to the α contours. In each case, the variable is largest near the wall and decreases towards the centre. In addition, at any given distance from the wall, the points near the diagonal axes of symmetry show a lower magnitude. The invariants η and ξ display nearly identical contours as expected from the points forming a nearly straight line of unit slope in [figure 8](#).

The area of lowest magnitude at the centre corresponds to the core flow, which is expected to be the region closest to isotropy, although the polar invariant map indicates the anisotropy magnitude remains approximately 15 % of the maximum value. The region of low magnitude along the symmetry planes is associated with the space between the corner vortices. This is an indication that the corner vortices are increasing the relative anisotropy magnitude.

The only variable that shows a significantly different distribution than the others is θ . In contrast to the cross-shape observed in the other contour maps, θ is only smaller along the axes of symmetry near the wall. Beyond the near-wall region, it increases nearly uniformly in a circular pattern, then decreases again in the core flow region. This reflects the movement of the anisotropy towards the plane strain line and then back towards the axisymmetric expansion line as seen in [figure 8](#).

The reason for the difference between the contours of θ and the other invariant map coordinates is due to the interdependence of the bounds on the coordinates in other invariant maps. For example, as η decreases, the range of ξ also decreases. Likewise, as C_3 increases, the upper limit on C_1 and C_2 decreases to satisfy the additivity property. Consequently, a correlation between the contours of each invariant coordinate is expected. In contrast, the range of acceptable values of α and θ never change, as any combination

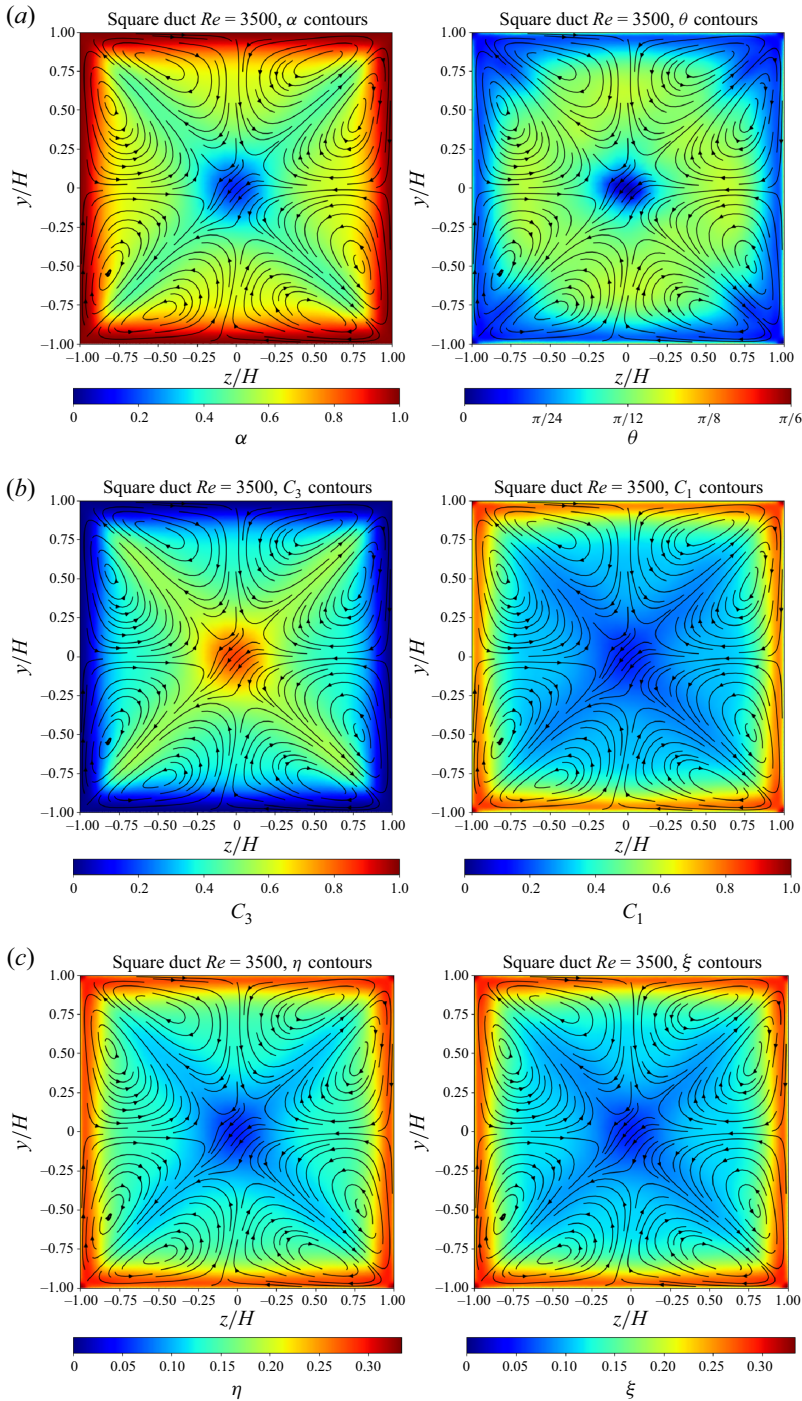


Figure 9. Contour maps of invariant coordinates overlaid with streamlines for the square duct case. The scales for θ and ξ have been reduced to half-range for clarity. (a) Polar map. left: α , right: θ , (b) Barycentric map. left: C_3 , right: C_1 , (c) ξ - η map. Left: η , Right: ξ .

of $\alpha \in [0, 1]$ and $\theta \in [0, \pi/3]$ is allowed. As a result, the θ contours show the similarity of the anisotropy along the symmetry lines, which is lost in the other representations.

5. Applications to machine learning-based turbulence modelling

This section will discuss the potential applications of the polar invariant map to turbulence modelling for realisability enforcement in comparison to other methods. It is divided into subsections discussing current approaches, the general strategy for usage of the polar invariant map, sample calculations with the polar invariant map and an example calculation using the k – ε model with a quadratic constitutive relation.

5.1. Current approaches to realisability

Most current approaches to realisability in ML turbulence modelling rely on the barycentric map, which is useful for interpolation between turbulent states due to its linearity (Banerjee *et al.* 2007). The most direct usage of the barycentric map is by applying a correction to a baseline RANS model in barycentric coordinates, then transforming the new normalised anisotropy tensor back to the original coordinate system, as done by Tracey *et al.* (2013) and Wang *et al.* (2017).

A comprehensive discussion of the workflow for constructing and implementing models based on the barycentric map was given by Wang *et al.* (2017) and Wu *et al.* (2018). The prediction process can be divided into several steps:

- (i) run a baseline RANS simulation to convergence;
- (ii) calculate eigenvalues and eigenvectors of the baseline solution;
- (iii) convert eigenvalues to the Cartesian coordinates on the barycentric mapping;
- (iv) use an ML model to predict the discrepancies in the eigenvalues, eigenvectors and turbulent kinetic energy, then add the discrepancies to the baseline solution;
- (v) apply the eigenvectors as transformation matrices to revert to standard coordinates.

Outside of running the baseline simulation, steps (ii) and (v) are the most expensive. The cost in step (ii) comes from performing the eigendecomposition of the flow field, while the cost in step (v) comes from the use of transformation matrices.

As a method for ensuring an ML model makes realisable predictions, the barycentric map is highly effective, but it has limited applicability to realisability enforcement outside of eigendecomposition-based ML turbulence modelling approaches. Another common approach to ML turbulence modelling is based on Pope's tensor basis formulation (Pope 1975). Using the Cayley–Hamilton theorem, it can be shown that all eddy viscosity and algebraic Reynolds stress models can be represented as the weighted sum of a finite number of tensors. The tensor basis is given by

$$b_{ij} = \sum_{n=1}^{10} G^{(n)} T^{(n)}, \quad (5.1)$$

where $T^{(n)}$ are basis tensors formed from the strain-rate and rotation-rate tensors, S_{ij} and W_{ij} , defined respectively as

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad W_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right). \quad (5.2)$$

Here, $G^{(n)}$ are scalar coefficients formed from the invariants of the basis tensors. The full list of definitions of tensors and invariants can be found from Pope (1975). Ling *et al.* (2016b) was the first to propose using machine learning to predict the coefficients of the tensor basis as a way to embed invariance properties. The approach, which they refer to as the tensor basis neural network (TBNN), and variants using other ML algorithms have been widely used in ML turbulence modelling research (Kaandorp & Dwight 2020; Mandler & Weigand 2022; Cai *et al.* 2024). In this case, there is no existing framework for creating models that inherently satisfy the realisability constraints, as purely invariant map-based approaches can. For tensor basis models, the need for baseline eigenvalues and eigenvectors limits current invariant maps to being used in post-processing corrections and training.

Blauw (2019) and Mandler & Weigand (2022) used the barycentric map to shift non-realistic predictions to the boundaries of the barycentric map in post-processing. Both works used the same process for points below the 1C–2C line:

- (i) calculate the eigendecomposition and convert to barycentric coordinates;
- (ii) if the point is non-realistic, shift it to the 1C–2C boundary;
- (iii) convert the new barycentric coordinates to eigenvalues.
- (iv) transform back to standard coordinates using the eigenvectors.

Mandler & Weigand (2022) also included additional corrections for points above the 1C–2C line on the barycentric map, which they found were necessary during the convergence process. As discussed in § 3.1, this only occurs when the predicted normalised anisotropy tensor is not traceless.

Blauw (2019) gives the new Cartesian coordinates on the barycentric map as

$$\begin{aligned} x_{B,c} &= x_B + \frac{x_{3c} - x_B}{y_{3c} - y_B}(y_{1c} - y_B), \\ y_{B,c} &= y_{1c}. \end{aligned} \quad (5.3)$$

Here, $x_{B,c}$ and $y_{B,c}$ are the corrected Cartesian coordinates on the barycentric map, and the original coordinates are x_B and y_B . The quantities x_{1c} , x_{3c} , y_{1c} and y_{3c} are the Cartesian coordinates of the corners of the barycentric map. This correction shifts the turbulent state along a line passing through the isotropic corner of the barycentric map.

Once the corrected coordinates are available, they are transformed to the corrected eigenvalues, $\lambda_{1,c} \geq \lambda_{2,c} \geq \lambda_{3,c}$, by solving the system of equations,

$$\begin{bmatrix} x_{1c} & (-x_{1c} + 2x_{2c}) & (-2x_{2c} + 3x_{3c}) \\ y_{1c} & (-y_{1c} + 2y_{2c}) & (-2y_{2c} + 3y_{3c}) \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_{1,c} \\ \lambda_{2,c} \\ \lambda_{3,c} \end{bmatrix} = \begin{bmatrix} x_{B,c} - x_{3c} \\ y_{B,c} - y_{3c} \\ 0 \end{bmatrix}. \quad (5.4)$$

Finally, the matrix of eigenvalues is transformed back to standard coordinates using

$$b_{ij,c} = \mathbf{V} \mathbf{\Lambda}_c \mathbf{V}^{-1}, \quad (5.5)$$

where $b_{ij,c}$ is the corrected anisotropy tensor, \mathbf{V} is the matrix of eigenvectors and $\mathbf{\Lambda}_c$ is the diagonal matrix of eigenvalues. This approach guarantees realisability, but does not affect the underlying model.

More recently, Riccius, Agrawal & Koutsourelakis (2023) and McConkey, Yee & Lien (2024) have proposed realisability-informed training, inspired by PINNs. Both added an additional term in the loss function based on realisability constraints to penalise non-realistic predictions. This approach explicitly ‘teaches’ the model realisability

constraints. As a representative example, McConkey *et al.* (2024) used the equations,

$$\begin{aligned} f_1 &= \max \left[(b_{ij} - 2/3), -(b_{ij} + 1/3), 0 \right], & i = j, \\ f_2 &= \max \left[(b_{ij} - 1/2), -(b_{ij} + 1/2), 0 \right], & i \neq j, \\ f_3 &= \max \left[(3|\lambda_2| - \lambda_2)/2 - \lambda_1, 0 \right], \\ f_4 &= \max \left[(\lambda_1 - (1/3 - \lambda_2)), 0 \right]. \end{aligned} \quad (5.6)$$

The first two equations give the violation of the realisability conditions on each component, while the second two give the violation of the eigenvalue conditions described by Schumann (1977). If a point is realisable, then the value is zero. These are combined in the penalty function,

$$\mathcal{L}_M = \gamma \left\{ \frac{1}{6} \left[\sum_{i=j} f_1^2 + \sum_{i \neq j} f_2^2 \right] + \frac{1}{2} [f_3^2 + f_4^2] \right\}. \quad (5.7)$$

Here, \mathcal{L}_M is the penalty function and γ is a tuning parameter controlling the strength of the penalty. Although it does not guarantee realisable predictions, McConkey *et al.* (2024) reported a substantial reduction in the number of non-realisable predictions. For example, they found that including the realisability-informed loss function reduced the number of non-realisable predictions in the square duct case from 9.2 % to only 0.4 %.

5.2. Polar invariant map strategies

The polar invariant map provides a framework that can simplify post-processing and realisability-informed training methods. Moreover, it can potentially enable the embedding of realisability constraints in tensor basis models. It allows realisability to be enforced through the use of α as a scaling factor, opening the door to new approaches. This is possible because of the representation of realisability as a limiting magnitude, which does not require an explicit eigendecomposition to calculate.

The simplest application for the polar invariant map is to shift non-realisable predictions to the nearest realisable state. In terms of the polar invariant map, the current method of shifting points to the 1C–2C boundary along a line connecting the isotropic corner is simply scaling along a line of constant θ . This is exactly the transformation described by

$$b_{ij,c} = \frac{1}{\alpha_0} b_{ij}, \quad (5.8)$$

where α_0 is the α value of b_{ij} . The only matrix operations required are a norm and a determinant for calculating α_0 . Removing the need for a cumbersome process of eigendecomposition, solving a linear system and applying rotation matrices can be expected to yield a substantial reduction in computation time. A comparison of the calculation time for the two methods is described in the following subsection.

The application of the polar invariant map for realisability-informed training is also straightforward. In current approaches, the penalty term is formulated using the component-wise constraints and the boundaries on the eigenvalues. The eigenvalue constraints require two equations to account for the upper and lower bounds. Using the polar invariant map, the eigenvalue constraints can be simplified to a single equation penalising $\alpha > 1$. Removing the need for an explicit eigenvalue calculation may yield substantial savings in training time over the thousands of iterations needed to train an ML model.

A more complex application is in turbulence models with adaptive coefficients. This avenue of research focuses on improving current turbulence models by using machine

learning to tune the coefficients based on the local flow physics. Most of these are based on a linear eddy viscosity model using the Boussinesq hypothesis, which posits a linear relationship to the velocity gradients of the form (Pope 2000)

$$\tau_{ij} = -2\nu_t S_{ij} + \frac{2}{3}k\delta_{ij} = a_{ij} + \frac{2}{3}k\delta_{ij}. \quad (5.9)$$

In this case, the anisotropy is simply a scaling of the strain-rate tensor by the eddy viscosity, ν_t . Because θ is independent of magnitude, the maximum anisotropy magnitude can be determined from the strain-rate tensor alone at every point in the flow field. Then, the limit on ν_t can be easily determined as the value that produces $\alpha = 1$,

$$\left| \frac{\nu_{t,max}}{k} \right| = \frac{1}{\alpha_S}. \quad (5.10)$$

Here, α_S is the value of α when S_{ij} is substituted for b_{ij} in (3.12). Having an inexpensive way to calculate the local maximum ν_t may be useful for actively constraining the model coefficients of adaptive turbulence models.

The knowledge of the local maximum ν_t and model definition could enable the development of adaptive models that can actively constrain the predicted coefficients based on local conditions.

The polar invariant map's potential applications are not limited to just simplifying existing approaches, but may also allow for incorporating realisability constraints into the structure of ML turbulence models. To see this, consider the output of an arbitrary turbulence model, $\mathcal{M}(\mathbf{x})$. Using the polar invariant map, the output can be separated into a tensor, $\tilde{\mathcal{M}}(\mathbf{x})$, with arbitrary magnitude, and a scaling in terms of α ,

$$b_{ij} = \mathcal{M}(\mathbf{x}) = \frac{\alpha_1}{\alpha_0} \tilde{\mathcal{M}}(\mathbf{x}). \quad (5.11)$$

Here, α_0 is the α value of $\mathcal{M}(\mathbf{x})$ and α_1 is the α value of b_{ij} . Equation (5.11) can be expanded to

$$b_{ij} = \frac{\alpha_1}{\sqrt{6}||\tilde{\mathcal{M}}(\mathbf{x})|| \cos\left(\theta - \frac{\pi}{3}\right)} \tilde{\mathcal{M}}(\mathbf{x}) = \frac{\alpha_1}{\sqrt{6} \cos\left(\theta - \frac{\pi}{3}\right)} \frac{\tilde{\mathcal{M}}(\mathbf{x})}{||\tilde{\mathcal{M}}(\mathbf{x})||}. \quad (5.12)$$

Finally, (5.12) may be simplified by recognising the denominator as $1/||b_{ij}||_{max}$ from (3.11) to the form

$$b_{ij} = \alpha_1 ||b_{ij}||_{max} \frac{\tilde{\mathcal{M}}(\mathbf{x})}{||\tilde{\mathcal{M}}(\mathbf{x})||}. \quad (5.13)$$

Equation (5.13) shows that the output is guaranteed to be realisable if and only if $\alpha_1 \in [0, 1]$. The practical consequence is that realisability can theoretically be incorporated into the structure of an arbitrary model by predicting $\alpha_1 \in [0, 1]$ in a second prediction step.

A general way to accomplish this is to extract the coefficient of (5.12) as a scaling factor of the form

$$F(\mathcal{M}(\mathbf{x}), \mathbf{y}) = \frac{\alpha_1(\mathbf{y})}{\sqrt{6}||\tilde{\mathcal{M}}(\mathbf{x})|| \cos\left(\theta - \frac{\pi}{3}\right)}. \quad (5.14)$$

Here, $F(\mathcal{M}(\mathbf{x}), \mathbf{y})$ is a function of the initial prediction $\mathcal{M}(\mathbf{x})$ and additional input variables \mathbf{y} . The inputs, \mathbf{y} , are used to predict α_1 , and $\mathcal{M}(\mathbf{x})$ is required to calculate θ and $||\mathcal{M}(\mathbf{x})||$.

Hypothetically, the above-mentioned strategy could be applied to any model architecture that can provide an initial prediction of the normalised anisotropy tensor with the correct θ . The resulting output will be realisable as long as $\alpha_1(\mathbf{y})$ remains between 0 and 1. Enforcing that range for $\alpha_1(\mathbf{y})$ can be easily accomplished by mapping the output to a sigmoid function – a common technique used in machine learning to predict classification probabilities. The usefulness of this approach would depend on the properties of $\mathcal{M}(\mathbf{x})$; for example, there would be no benefit to barycentric map-based models, which already enforce realisability. In contrast, it could be highly useful as a way to incorporate realisability constraints in the structure of a broad range of tensor basis models. The full range of benefits, drawbacks and details of implementation for such an approach remains a subject for further research.

5.2.1. Numerical considerations

All of the previously discussed potential applications for the invariant map are based on the assumption that θ is calculated from a symmetric, traceless tensor. These assumptions are based on the properties of the normalised anisotropy tensor, which, by definition, is symmetric and traceless. However, in practice, the traceless property of b_{ij} may not be satisfied while a CFD solution is converging, leading to violations of the domain of the arccosine in calculating θ . It is therefore important to consider how to handle these cases.

The reason for the violation of the traceless property of b_{ij} can be easily seen by considering a linear eddy viscosity model,

$$b_{ij} = \frac{a_{ij}}{2k} = -\frac{\nu_t}{k} S_{ij}. \quad (5.15)$$

It is clear from (5.15) that an eddy viscosity model will only predict a traceless b_{ij} if S_{ij} is also traceless. Physically, this is guaranteed in an incompressible flow because the trace of S_{ij} is identically zero as a consequence of the conservation of mass (White & Majdalani 2021). In a solver, however, there is no guarantee that this will be the case before a solution converges. In fact, violations of the conservation of mass are frequently used as a convergence criterion for this reason. If conservation of mass is not satisfied at a point, then b_{ij} will incorrectly have a trace. This reasoning also extends to the full tensor basis, which includes S_{ij} as the first basis tensor.

The simplest method of handling non-traceless input tensors with the polar invariant map is to use a modified strain-rate tensor formed by removing the trace,

$$S_{ij}^* = S_{ij} - \frac{1}{3} S_{kk} \delta_{ij}. \quad (5.16)$$

Equation (5.16) only impacts the solution when a converging solution violates the conservation of mass. As the solution converges, S_{ij}^* converges to S_{ij} . By using S_{ij}^* to calculate the basis tensors, the zero trace property of b_{ij} can be ensured throughout the entire convergence process.

It is interesting to note that S_{ij}^* is mathematically identical to the traceless strain-rate tensor used by eddy viscosity models in compressible flows (Wilcox 2006). Although S_{ij}^* is motivated by numerical artefacts, unlike the compressible flow case, where it is necessary for physical reasons, the same constitutive relation is successfully used by general-purpose CFD codes throughout both regimes (Pletcher, Tannehill & Anderson 2016). This may be taken as an indication that S_{ij}^* is a numerically feasible solution to the problem of non-traceless input tensors.

5.3. Post-processing corrections: sample calculations

We will now demonstrate the potential of the polar invariant for simplifying and reducing the time for calculations by making a comparison between the correction method of (5.8) and the formulation of Blauw (2019). The comparison was conducted in Python on randomly generated, non-realisable, normalised anisotropy tensors. The methodology can be divided into four steps:

- (i) randomly generate a set of N symmetric, deviatoric matrices;
- (ii) apply both correction methods sequentially to each point;
- (iii) record the total calculation time;
- (iv) repeat the calculations on the same set M times and average the calculation times.

To calculate the percent reduction in calculation time, the following formula was used:

$$\text{Percent Reduction in Time} = 100 \cdot \left(\frac{T_{\text{bary}} - T_{\text{polar}}}{T_{\text{bary}}} \right), \quad (5.17)$$

where T_{bary} is the time to correct N tensors using the barycentric formulation and T_{polar} is the time to correct N tensors using the polar formulation.

For the first step, the Numpy *default_rng* and *random* functions were used to generate random 3×3 matrices. These matrices were made symmetric by adding the transpose and dividing by two. They were then made deviatoric by subtracting $1/3$ of the trace from each element on the diagonal. This is depicted in Algorithm 2 in Appendix B.

To implement the barycentric corrections, the Numpy *eigh* function was used. The *eigh* function uses the LAPACK *_syevd* routine to calculate the eigenvalues and eigenvectors of a real, symmetric matrix. The eigenvalues were converted to barycentric coordinates according to (2.9). The barycentric coordinates were converted to Cartesian form via (4.6). Once the Cartesian coordinates on the barycentric map were obtained, the correction was calculated with (5.3), converted to eigenvalues through (5.4) and transformed back to standard coordinates via (5.5). All of these steps were combined into a single function. The pseudo-code for the barycentric correction is shown in Algorithm 3 in Appendix B.

The polar invariant corrections were implemented using (3.6b) and (3.12) to calculate α_0 . The norm of b_{ij} was calculated using the Numpy *norm* function. Once α_0 was obtained, the correction was applied with (5.8). These steps were combined into a single function. The pseudo-code for the polar correction is shown in Algorithm 4 in Appendix B.

For the test, $N = 10\,000$ random matrices were generated with a random seed of 0. The corrections were applied sequentially using separate for loops. The time required for each function evaluation was recorded using the Python *time* function in the *time* module. The percent difference in the total time relative to the barycentric formulation was calculated according to (5.17). To reduce the effect of random variations in the performance of the processor, an outer for loop was used to repeat these calculations $M = 1000$ times. The percent differences for each run were recorded and averaged to produce the final time estimate. All calculations were performed on an Intel i9–13980HX processor.

Figure 10 shows the distribution of percent reduction in calculation time over the 1000 runs, and compares the initial and corrected points in the polar invariant map for a sample run. The histogram in figure 10(a) shows that the reduction in calculation time across the runs forms an approximately normal distribution. This is the expected result if there are no systemic biases in the variation of processor performance over time. Figure 10(b) confirms that the polar formulation correctly shifts all points to the 1C–2C line. Overall, the results

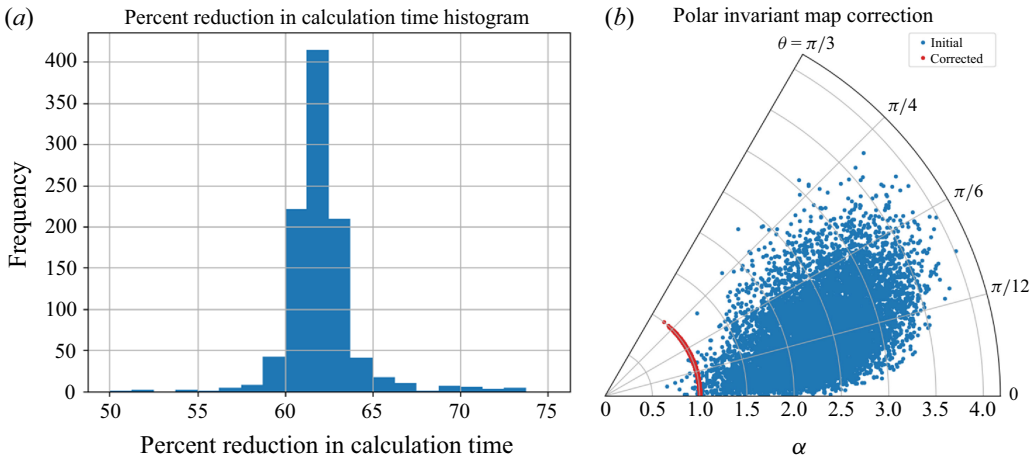


Figure 10. (a) A histogram of the percent reduction in computation time of the polar invariant formulation compared with the barycentric map formulation. The mean is 62 %. (b) A comparison of the initial and corrected points in the polar invariant map for one set of 10 000 random matrices.

of the test show the polar formulation is equivalent to the barycentric formulation, but with an average reduction in calculation time of 62 %.

5.4. Coefficient constraints: sample calculations

To illustrate how α – θ coordinates can be used to find the limiting magnitude of turbulence model coefficients, consider the simple case of the quadratic constitutive relation (QCR2000) proposed by Spalart (2000),

$$\tau_{ij,QCR} = 2v_t S_{ij} - C_{cr1}(O_{ik}\tau_{kj} - \tau_{ik}O_{kj}), \quad (5.18a)$$

$$O_{ik} = \frac{2W_{ik}}{\sqrt{\frac{\partial u_m}{\partial x_n} \frac{\partial u_m}{\partial x_n}}}. \quad (5.18b)$$

Here, W_{ik} is the rate of rotation tensor, v_t is the linear eddy viscosity and C_{cr1} is a constant equal to 0.3. The term $\sqrt{(\partial u_m / \partial x_n)(\partial u_m / \partial x_n)}$ is the velocity gradient magnitude, where u_m is the mean velocity. For brevity, it will be represented as $||\nabla U||$ from this point forward. As an example of how the polar invariant map can be applied to tensor basis models, we will demonstrate how it may be used to derive a limit on the eddy viscosity and C_μ in the k – ε turbulence model.

In the case of QCR2000, the definition of O_{ik} can be substituted back into (5.18a) to convert it to the form

$$a_{ij} = 2v_t S_{ij} + \frac{4v_t C_{cr1}}{||\nabla U||} (S_{ik} W_{kj} - W_{ik} S_{kj}). \quad (5.19)$$

Note that the contribution to τ_{ij} from the turbulent kinetic energy can be added afterwards due to the anti-symmetric tensor in the equation (Spalart 2000). Normalising by $2k$ and factoring out v_t/k yields

$$b_{ij} = \frac{v_t}{k} \left[S_{ij} + \frac{2C_{cr1}}{||\nabla U||} (S_{ik} W_{kj} - W_{ik} S_{kj}) \right]. \quad (5.20)$$

The terms in parentheses correspond to $T^{(n)}$ from (5.1). Specifically, $S_{ij} = T^{(1)}$ and $(S_{ik}W_{kj} - W_{ik}S_{kj}) = T^{(2)}$ are the first two basis tensors of Pope's general eddy viscosity formulation (Pope 1975).

Comparing (5.20) to (5.11), we can identify ν_t/k with α_1/α_0 and the term in brackets with $\tilde{\mathcal{M}}(\mathbf{x})$. Setting these equal yields

$$\frac{\nu_t}{k} = \frac{\alpha_1}{\sqrt{6} \|\tilde{\mathcal{M}}(\mathbf{x})\| \cos\left(\theta - \frac{\pi}{3}\right)}. \quad (5.21)$$

The maximum realisable eddy viscosity occurs when $\alpha_1 = 1$, so the limit on the eddy viscosity is

$$\left| \frac{\nu_t}{k} \right| \leq \frac{1}{\sqrt{6} \|T^{(1)} + \frac{2C_{cr1}}{\|\nabla U\|} T^{(2)}\| \cos\left(\theta - \frac{\pi}{3}\right)}. \quad (5.22)$$

This is an equation that can be solved inexpensively and analytically for any $T^{(1)}$ and $T^{(2)}$. With the local maximum ν_t available, it becomes possible to use a model that predicts ν_t relative to the maximum realisable magnitude.

Next, consider an eddy viscosity as prescribed by the k - ε model (White & Majdalani 2021),

$$\nu_t = C_\mu \frac{k^2}{\varepsilon}. \quad (5.23)$$

Here, C_μ is a proportionality constant. Substituting (5.23) into (5.22) allows a limit to be placed on C_μ according to

$$|C_\mu| \leq \frac{\sqrt{6}}{6(k/\varepsilon) \|T^{(1)} + \frac{2C_{cr1}}{\|\nabla U\|} T^{(2)}\| \cos\left(\theta - \frac{\pi}{3}\right)}. \quad (5.24)$$

Equation (5.24) relates the local maximum realisable magnitude to the proportionality coefficient in the k - ε model. This form of the limiting magnitude would be useful to a turbulence model with adaptive coefficients. Additional research would be necessary to determine how other coefficients could be related to the limiting magnitude.

Consider the case of a high-Reynolds-number turbulent round jet as an example of the limitation on C_μ . The flow is axisymmetric and non-swirling, so the only non-zero mean velocity components are the axial velocity, u , and the radial velocity, v . By order of magnitude arguments, $v \ll u$ and $\partial/\partial x \ll \partial/\partial r$ (Pope 2000). Then, for (5.19), S_{ij} and W_{ij} can be approximated as having only the components $S_{12} = S_{21} = u_r/2$ and $W_{12} = -W_{21} = u_r/2$. In this case, $T^{(1)}$ and $T^{(2)}$ are

$$T^{(1)} = \begin{bmatrix} 0 & u_r/2 & 0 \\ u_r/2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad T^{(2)} = \begin{bmatrix} -u_r^2/2 & 0 & 0 \\ 0 & u_r^2/2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (5.25)$$

The velocity gradient magnitude is $|u_r|$, so T_{ij} becomes

$$T_{ij} = \left(T^{(1)} + \frac{2C_{cr1}}{|u_r|} T^{(2)} \right) = \begin{bmatrix} -(u_r^2/|u_r|) C_{cr1} & u_r/2 & 0 \\ u_r/2 & (u_r^2/|u_r|) C_{cr1} & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (5.26)$$

Two simplifications can be made to (5.26). First, $u_r^2/|u_r|$ is strictly non-negative and can simply be replaced with $|u_r|$. Additionally, it is well known that u_r is strictly non-positive away from the edge of a round jet, so in this specific case, $-u_r$ is strictly positive, so $|u_r| = -u_r$. Making these simplifications yields

$$T_{ij} = \left(T^{(1)} + \frac{2C_{cr1}}{|u_r|} T^{(2)} \right) = \begin{bmatrix} u_r C_{cr1} & u_r/2 & 0 \\ u_r/2 & -u_r C_{cr1} & 0 \\ 0 & 0 & 0 \end{bmatrix} = u_r \begin{bmatrix} 3/10 & 1/2 & 0 \\ 1/2 & -3/10 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (5.27)$$

The single zero-column indicates this is a case of plane-strain, so $\theta = \pi/6$ and $\cos(\theta - \pi/3) = \sqrt{3}/2$. The magnitude of T_{ij} is $\sqrt{17}/5$. Substituting these values into (5.22) gives a limit on the eddy viscosity of

$$\left| \frac{\nu_t}{k} \right| \leq \frac{5\sqrt{34}}{51|u_r|} \approx \frac{0.572}{|u_r|}. \quad (5.28)$$

For the $k-\varepsilon$ model, this implies the limit on C_μ in a round jet using the quadratic constitutive relation is

$$|C_\mu| \leq \frac{5\sqrt{34}}{51|u_r|(k/\varepsilon)} \approx \frac{0.572}{|u_r|(k/\varepsilon)}. \quad (5.29)$$

In general, (5.21) must be solved at each point in the flow field due to differences in the relative weights, $G^{(n)}$, of (5.1). The above-mentioned example demonstrates the type of analysis involved in applying the polar invariant map to non-linear eddy viscosity models.

6. Conclusion

This paper introduces a new polar turbulent invariant map based on an angle, θ , which characterises the relative values of principal anisotropic stresses, and a scalar describing the distance from isotropy, α . The polar invariant map complements prior works by reframing realisability as a limit on the anisotropy magnitude. It is shown that the new mapping has the properties that all non-realizable states have $\alpha > 1$ and that the nearest realizable point has the same θ but with $\alpha = 1$. These properties allow $1/\alpha$ to be applied directly to a predicted Reynolds stress tensor as a scaling factor. Consequently, non-realizable predictions can be brought to the nearest realizable state without the need for explicit calculation of eigenvalues or coordinate transformations.

Mathematical relationships between the polar invariant map, the $\xi-\eta$ invariant map and the barycentric map are developed. It is shown that the barycentric map compresses the θ isolines as the two-component corner is approached, while the $\xi-\eta$ invariant map expands the θ isolines near the plain-strain limit and compresses them everywhere else. The polar invariant map is found to provide an undistorted representation of the relationship between eigenvalues, while mildly compressing the one-component corner relative to the barycentric map.

Comparisons between the polar invariant map and other invariant maps are made for a fully developed channel flow profile and a square duct cross-section. It is shown that the barycentric and polar invariants generally provide similar visualisations of the turbulent state, while the $\xi-\eta$ invariant map tends to obscure the trends in comparison. The polar

invariant map is seen to quantify differences in the principal stresses and anisotropy relative to its limiting magnitude that are not easily obtained from the barycentric and ξ – η maps. Contour maps of each invariant coordinate are displayed for the square duct case. The distribution of θ demonstrates that the reduction in ξ , η , C_1 and the increase in C_3 are primarily due to scaling of the anisotropy tensor instead of a change in the ratio of principal anisotropic stresses. This is an observation not readily apparent in the other invariant maps due to the interdependence of the bounds on each coordinate.

The applications of the polar invariant map to machine learning-based turbulence modelling are discussed. Potential applications include simplifying post-processing realisability corrections and realisability-informed training of ML models. More complex applications are constraining the coefficients of adaptive turbulence models and general nonlinear eddy viscosity models. An analytical example is shown for how the polar invariant map can be used to constrain model coefficients using the k – ε model in a free shear flow with Spalart’s quadratic constitutive relation. A comparison of the barycentric and equivalent polar formulation of realisability corrections showed an average 62 % decrease in calculation time. This serves as a representative example of the advantages of removing the need for explicit eigendecomposition and transformation matrices for enforcing realisability in ML turbulence modelling.

In conclusion, the polar invariant map can serve as a foundation for new approaches to realisable machine learning-based turbulence modelling. The reframing of realisability in terms of a scaling factor provides a flexible framework for diverse machine-learning models. Future work can build off of this research by investigating the challenges in application to specific methodologies. For example, there is clearly a strong relationship between wall distance and α ; however, it remains to be seen what other variables may be useful to its prediction. Work is also needed to determine the specific modifications necessary to integrate it with different approaches, say, for adaptive turbulence models versus tensor basis expansion approaches. The authors are currently investigating modifications to the tensor basis neural network using the polar invariant map as a foundation.

Acknowledgements. The authors would like to thank the reviewers, whose insightful comments greatly improved the quality of this article. This work was sponsored by the Air Force Office of Scientific Research Computational Mathematics Program under Program Officer Dr. Fariba Fahroo. This is a work of the US Government and is not subject to copyright protection in the United States.

Funding. This material is based on research sponsored by AFRL under agreement number FA8650-24-2-9350. The U.S. Government is authorised to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

The opinions, findings, views, conclusions or recommendations contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the DAF, AFRL or the U.S. Government.

Declaration of interests. The authors report no conflict of interest.

Data availability statement. Distribution Statement A: Approved for public release; distribution is unlimited. PA# AFRL-2024-5752.

Appendix A.

A proof that (3.6b) has a solution for all symmetric, deviatoric tensors is described here. The domain of the arccosine is $[-1,1]$, so it must be shown that for any symmetric, deviatoric tensor, b_{ij} ,

$$\left| 3\sqrt{6} \det \left(\frac{b_{ij}}{\|b_{ij}\|} \right) \right| \leq 1. \quad (\text{A1})$$

Because the determinant is a tensor invariant, (A1) may be rewritten in terms of the eigenvalues of b_{ij} as (Pope 2000)

$$\left| 3\sqrt{6} \frac{\lambda_1 \lambda_2 \lambda_3}{(\lambda_1^2 + \lambda_2^2 + \lambda_3^2)^{3/2}} \right| \leq 1. \quad (\text{A2})$$

The trace of a deviatoric tensor is zero, so $\lambda_3 = -\lambda_1 - \lambda_2$. Making this substitution and simplifying yields

$$\left| 3\sqrt{6} \frac{\lambda_1^2 \lambda_2 + \lambda_1 \lambda_2^2}{(2\lambda_1^2 + 2\lambda_1 \lambda_2 + 2\lambda_2^2)^{3/2}} \right| \leq 1. \quad (\text{A3})$$

Equation (A3) can be further simplified by factoring out the two and replacing the absolute value with the square of both sides,

$$\left(\frac{3\sqrt{3}}{2} \frac{\lambda_1^2 \lambda_2 + \lambda_1 \lambda_2^2}{(\lambda_1^2 + \lambda_1 \lambda_2 + \lambda_2^2)^{3/2}} \right)^2 \leq 1^2, \quad (\text{A4})$$

$$\frac{27}{4} \frac{(\lambda_1^2 \lambda_2 + \lambda_1 \lambda_2^2)^2}{(\lambda_1^2 + \lambda_1 \lambda_2 + \lambda_2^2)^3} \leq 1. \quad (\text{A5})$$

This is valid provided that the argument of the absolute value is a real number. This is guaranteed to be satisfied for b_{ij} because a symmetric tensor has all real eigenvalues (Chong & Zak 2013). Then to prove the inequality holds, it is sufficient to show that it holds for the maximum value of

$$f(\lambda_1, \lambda_2) = \frac{(\lambda_1^2 \lambda_2 + \lambda_1 \lambda_2^2)^2}{(\lambda_1^2 + \lambda_1 \lambda_2 + \lambda_2^2)^3}. \quad (\text{A6})$$

Finding the global maximum of (A6) can be accomplished more easily by rewriting (A6) in polar coordinates where $\lambda_1 = \rho \cos \phi$, $\lambda_2 = \rho \sin \phi$, $\rho = \sqrt{\lambda_1^2 + \lambda_2^2}$ and $\phi = \arctan(\lambda_2/\lambda_1)$. Making these substitutions gives

$$f(\rho, \phi) = \frac{(\rho^3 \cos^2 \phi \sin \phi + \rho^3 \cos \phi \sin^2 \phi)^2}{(\rho^2 \cos^2 \phi + \rho^2 \cos \phi \sin \phi + \rho^2 \sin^2 \phi)^3}. \quad (\text{A7})$$

In (A7), ρ cancels out, so it becomes a single variable function of ϕ ,

$$f(\phi) = \frac{(\cos^2 \phi \sin \phi + \cos \phi \sin^2 \phi)^2}{(\cos^2 \phi + \cos \phi \sin \phi + \sin^2 \phi)^3}. \quad (\text{A8})$$

Equation (A8) can be simplified using the trigonometric identities $\sin^2 \phi + \cos^2 \phi = 1$ and $\sin 2\phi = 2 \sin \phi \cos \phi$. This yields

$$f(\phi) = \frac{\left(\frac{1}{2} \sin 2\phi (\cos \phi + \sin \phi) \right)^2}{\left(1 + \frac{1}{2} \sin 2\phi \right)^3}. \quad (\text{A9})$$

Factoring out the constants then gives

$$f(\phi) = \frac{2(\sin 2\phi(\cos \phi + \sin \phi))^2}{(2 + \sin 2\phi)^3}. \quad (\text{A10})$$

Finally, expanding the numerator results in

$$f(\phi) = \frac{2(1 + \sin 2\phi) \sin^2 2\phi}{(2 + \sin 2\phi)^3}. \quad (\text{A11})$$

Equation (A11) is periodic with a period of π radians. Consequently, the global maximum of $f(\lambda_1, \lambda_2)$ can be determined by finding the maximum of $f(\phi)$ over the closed interval $[-(\pi/2), (\pi/2)]$ (Chong & Zak 2013). This can be accomplished by setting the derivative of $f(\theta)$ equal to zero and evaluating each critical point.

The derivative of $f(\phi)$ is

$$\frac{df}{d\theta} = \frac{4 \cos 2\phi \sin 2\phi (5 \sin 2\phi + 4)}{(\sin 2\phi + 2)^4}. \quad (\text{A12})$$

The unique roots of this equation are solutions to $\cos 2\phi = 0$, $\sin 2\phi = 0$ or $(5 \sin 2\phi + 4) = 0$ over the interval $[-(\pi/2), (\pi/2)]$. These solutions are

$$\theta = \pm \frac{\pi}{4}, \quad \theta = \pm \frac{\pi}{2}, \quad \theta = -\frac{1}{2} \arcsin\left(\frac{4}{5}\right), \quad \theta = \frac{1}{2} \arcsin\left(\frac{4}{5}\right) - \frac{\pi}{2}. \quad (\text{A13})$$

Evaluating each critical point gives the local extrema as

$$\begin{aligned} f\left(-\frac{\pi}{4}\right) &= f\left(-\frac{\pi}{2}\right) = f\left(\frac{\pi}{2}\right) = 0, \\ f\left(\frac{\pi}{4}\right) &= f\left(-\frac{1}{2} \arcsin\left(\frac{4}{5}\right)\right) = f\left(\frac{1}{2} \arcsin\left(\frac{4}{5}\right) - \frac{\pi}{2}\right) = \frac{4}{27}. \end{aligned} \quad (\text{A14})$$

Therefore, the maximum value of $f(\theta)$ is $4/27$. Substituting the maximum value of $f(\theta)$ into (A5) completes the proof:

$$\frac{27}{4} \times \frac{4}{27} = 1 \leq 1. \quad (\text{A15})$$

Thus, it is proven that (3.6b) has a solution for all symmetric, deviatoric tensors.

Appendix B.

Algorithm 1 Colouring invariant maps using barycentric coordinates

Require: Invariant map data containing barycentric coordinates.

function COLORINVARIANTMAPS(invariantMapData)

$pixelColors \leftarrow$ empty list

for (C_1, C_2, C_3) in *invariantMapData* **do**

$RGB_triplet \leftarrow (C_1, C_2, C_3)$

 Append $RGB_triplet$ to $pixelColors$

end for

return $pixelColors$

 ▷ List of RGB triplets for pixel colouring

end function

$barycentricColorList \leftarrow$ COLORSCATTERPLOT(*invariantMapData*)

 ▷ Use *barycentricColorList* as RGB values to colour the scatter plot pixels.

Algorithm 2 Generating random, symmetric, deviatoric 3×3 matrices

Require: Randomly generated stack of 3×3 matrices.
function RANDOMANISOTROPYTENSOR(randomMatrix)
 randomTensor \leftarrow empty list
 for *A* in *RandomMatrix* **do**
 $A_{\text{symm}} \leftarrow \frac{1}{2}(A + A^T)$
 $A_{\text{symm},\text{dev}} \leftarrow A_{\text{symm}} - \frac{1}{3}\mathbf{I} \cdot \{A_{\text{symm}}\}$
 Append $A_{\text{symm},\text{dev}}$ to *randomTensor*
 end for
 return *randomTensor*
end function

Algorithm 3 Barycentric map-based correction

Require: Initial prediction of b_{ij} .
function BARYCENTRICCORRECTION(b_{ij})
 Compute eigenvalues $\lambda_1, \lambda_2, \lambda_3$, and eigenvector matrix V
 $C_1 \leftarrow \lambda_1 - \lambda_2$
 $C_2 \leftarrow 2(\lambda_2 - \lambda_3)$
 $C_3 \leftarrow 1 + 3\lambda_3$
 $x_B \leftarrow C_1 + \frac{1}{2}C_3$
 $y_B \leftarrow \frac{\sqrt{3}}{2}C_3$
 $x_{B,c} \leftarrow x_B - \frac{1/2 - x_B}{\sqrt{3}/2 - y_B} y_B$
 $y_{B,c} \leftarrow 0$
 Solve $\begin{bmatrix} 1 & -1 & -\frac{3}{2} \\ 0 & 0 & \frac{3\sqrt{3}}{2} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_{1,c} \\ \lambda_{2,c} \\ \lambda_{3,c} \end{bmatrix} = \begin{bmatrix} x_{B,c} - \frac{1}{2} \\ y_{B,c} - \frac{\sqrt{3}}{2} \\ 0 \end{bmatrix}$
 $A_c \leftarrow \begin{bmatrix} \lambda_{1,c} & 0 & 0 \\ 0 & \lambda_{2,c} & 0 \\ 0 & 0 & \lambda_{3,c} \end{bmatrix}$
 $b_{ij,c} \leftarrow V A_c V^{-1}$
 return $b_{ij,c}$
end function

Algorithm 4 Polar invariant map-based correction

Require: Initial prediction of b_{ij} .
function POLARCORRECTION(b_{ij})
 $\theta \leftarrow \frac{1}{3} \arccos \left(3\sqrt{6} \det \left(\frac{b_{ij}}{\|b_{ij}\|} \right) \right)$
 $\alpha_0 \leftarrow \sqrt{6} \|b_{ij}\| \cos \left(\theta - \frac{\pi}{3} \right)$
 $b_{ij,c} \leftarrow \frac{1}{\alpha_0} b_{ij}$
return $b_{ij,c}$
end function

REFERENCES

- ANTONIA, R.A., KIM, J. & BROWNE, L.W.B. 1991 Some characteristics of small-scale turbulence in a turbulent duct flow. *J. Fluid Mech.* **233**, 369–388.
- BAKER, A.C. & PORTEOUS, H.L. 1990 *Linear Algebra and Differential Equations*. Ellis Horwood Ltd.
- BANERJEE, S., KRAHL, R., DURST, F. & ZENGER, CH 2007 Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches. *J. Turbul.* **8**, N32.
- BLAUW, C.A. 2019 Bayesian additive regression trees for data-driven rans turbulence modelling. Master's thesis, TU Delft, The Netherlands.
- CAI, J., ANGELI, P.-E., MARTINEZ, J.-M., DAMBLIN, G. & LUCOR, D. 2024 Revisiting tensor basis neural network for Reynolds stress modeling: application to plane channel and square duct flows. *Comput. Fluids* **275**, 106246.
- CHAKRABARTY, J. 2006 Chapter one – stresses and strains. In *Theory of Plasticity*. 3rd edn, (ed. J. Chakrabarty), pp. 1–55. Butterworth–Heinemann.
- CHOI, K.-S. & LUMLEY, J.L. 2001 The return to isotropy of homogeneous turbulence. *J. Fluid Mech.* **436**, 59–84.
- CHONG, E.K.P. & ZAK, S.H. 2013 *Elements of Statistical Learning*. 4th edn. John Wiley & Sons Inc.
- DURASAMY, K., IACCARINO, G. & XIAO, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51** (2019), 357–377.
- EMORY, M. & IACCARINO, G. 2014 Visualizing turbulence anisotropy in the spatial domain with componentiality contours. In *Center for Turbulence Research Annual Research Briefs*, pp. 123–138. Stanford University.
- EMORY, M., LARSSON, J. & IACCARINO, G. 2013 Modeling of structural uncertainties in Reynolds-averaged Navier–Stokes closures. *Phys. Fluids* **25** (11), 110822.
- FRANCESCHINI, L., SIPP, D. & MARQUET, O. 2020 Mean-flow data assimilation based on minimal correction of turbulence models: application to turbulent high Reynolds number backward-facing step. *Phys. Rev. Fluids* **5**, 094603.
- HOYAS, S. & JIMÉNEZ, J. 2006 Scaling of the velocity fluctuations in turbulent channels up to $Re = 2003$. *Phys. Fluids* **18** (1), 011702.
- IRGENS, F. 2019 *Tensor Analysis*. 1st edn. Springer Nature.
- KAANDORP, M.L.A. & DWIGHT, R.P. 2020 Data-driven modelling of the Reynolds stress tensor using random forests with invariance. *Comput. Fluids* **202**, 104497.
- KASSIR, M. 2017 *Applied Elasticity and Plasticity*. CRC Press.
- KROGSTAD, P.-Å. & TORBERGSEN, L.E. 2000 Invariant analysis of turbulent pipe flow. *Flow Turbul. Combust.* **64**, 161–181.
- LING, J., JONES, R. & TEMPLETON, J. 2016a Machine learning strategies for systems with invariance properties. *J. Comput. Phys.* **318**, 22–35.
- LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016b Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166.
- LUMLEY, J.L. 1978 Computational modeling of turbulent flows. *Adv. Appl. Mech.* **18**, 123–176.
- LUMLEY, J.L. & NEWMAN, G.R. 1977 The return to isotropy of homogeneous turbulence. *J. Fluid Mech.* **82** (1), 161–178.
- MANDLER, H. & WEIGAND, B. 2022 A realizable and scale-consistent data-driven non-linear eddy viscosity modeling framework for arbitrary regression algorithms. *Intl J. Heat Fluid Flow* **97**, 109018.
- MCCONKEY, R., YEE, E. & LIEN, F.-S. 2021 A curated dataset for data-driven turbulence modelling. *Sci. Data* **8** (1), 255.
- MCCONKEY, R., YEE, E. & LIEN, F.-S. 2024 Realizability-informed machine learning for turbulence anisotropy mappings. [arXiv:2406.11603](https://arxiv.org/abs/2406.11603)

- PINELLI, A., UHLMANN, M., SEKIMOTO, A. & KAWAHARA, G. 2010 Reynolds number dependence of mean flow structure in square duct turbulence. *J. Fluid Mech.* **644**, 107–122.
- PLETCHER, R.H., TANNEHILL, J.C. & ANDERSON, D.A. 2016 *Computational Fluid Mechanics and Heat Transfer*, 3rd edn. CRC Press.
- POPE, S.B. 1975 A more general effective-viscosity hypothesis. *J. Fluid Mech.* **72** (2), 331–340.
- POPE, S.B. 2000 *Turbulent Flows*. Cambridge University Press.
- RICCIUS, L., AGRAWAL, A. & KOUTSOURELAKIS, P.-S. 2023 Physics-informed tensor basis neural network for turbulence closure modeling. arXiv: 2311.14576.
- ROTTA, J. 1951 Statistische theorie nichthomogener turbulenz. *Z. Phys.* **129** (6), 547–572.
- SCHUMANN, U. 1977 Realizability of Reynolds-stress turbulence models. *Phys. Fluids* **20** (5), 721–725.
- SIMONSEN, A.J. & KROGSTAD, P.-Å. 2005 Turbulent stress invariant analysis: clarification of existing terminology. *Phys. Fluids* **17** (8), 088103.
- SPALART, P.R. 2000 Strategies for turbulence modelling and simulations. *Intl J. Heat Fluid Flow* **21** (3), 252–263.
- TRACEY, B., DURAISAMY, K. & ALONSO, J. 2013 Application of supervised learning to quantify uncertainties in turbulence and combustion modeling. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*.
- WANG, J.-X., WU, J.-L. & XIAO, H. 2017 Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on dns data. *Phys. Rev. Fluids* **2**, 034603.
- WHITE, F.M. & MAJDALANI, J. 2021 *Viscous Fluid Flow*, 4th edn. McGraw-Hill Education.
- WILCOX, D. 2006 *Turbulence Modeling for CFD*, 3rd edn. DCW Industries.
- WU, J.-L., XIAO, H. & PATERSON, E. 2018 Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework. *Phys. Rev. Fluids* **3**, 074602.
- ZUCKER, I.J. 2008 The cubic equation – a new look at the irreducible case. *Math. Gazette* **92** (524), 264–268.