

# Parallel QR algorithm for data-driven decompositions

By T. Sayadi<sup>†</sup>, C. W. Hamman AND P. J. Schmid<sup>‡</sup>

Many fluid flows of engineering applications, although very complex in appearance, can be approximated by lower-order models governed by a few modes, able to capture the dominant behavior (dynamics) of the system. Recently, different techniques have been developed, designed to extract the most dominant coherent structures from the flow. Some of the more general techniques are based on data-driven decompositions, most of which rely on performing a singular value decomposition (SVD) on a formulated snapshot matrix. As the number of degrees of freedom of a simulation increases, the resulting data-matrix becomes longer, otherwise referred to as a tall-and-skinny (TS) matrix. Ultimately, the SVD of a TS data-matrix can no longer be handled on a single processor. To overcome this limitation, the present study employs the parallel TSQR algorithm of Demmel *et al.* (2012), which is further used as a basis of the underlying parallel SVD. This algorithm is shown to scale well on machines with a large number of processors and, therefore, allows the decomposition of very large data-sets.

---

## 1. Background and motivation

Data-driven decomposition techniques are employed to extract coherent structures from flows of engineering applications. Since these decomposition techniques rely solely on the underlying data-set, they are applicable to numerical or experimental results alike, making them useful and versatile tools for post-processing purposes. Proper orthogonal decomposition (POD), for example, belongs to this class of methods, where structures are ranked by their energy content and form an orthogonal basis of the analyzed data-set. Often, more than one frequency is attributed to each POD mode, since the optimization process is focused on the spatial content of the data rather than on the time evolution. Balanced POD (Rowley 2005) is an alternative to POD for input-output systems and constitutes a tractable method for computing approximate balanced truncations of large-scale state-space systems used in control applications. Another example of a data-decomposition technique is Koopman (Mezic 2013, Rowley *et al.* 2009) or dynamic (Schmid 2010) mode decomposition. The result of this decomposition algorithm is a non-orthogonal basis, where each mode is associated with a specific frequency present throughout the data. DMD has recently been applied to various flow configurations yielding interesting results (Rowley *et al.* 2009, Grilli *et al.* 2012, Sayadi *et al.* 2014).

All of these data-decomposition techniques, however, rely on an SVD of the data-matrix. As the size of the analyzed data increases, through highly resolved simulations or better-quality experimental measurements, the number of rows of the snapshot-matrix increases, resulting in a tall-and-skinny (TS) snapshot-matrix. The TS data matrix will

<sup>†</sup> Aerospace Engineering Department, University of Illinois, Urbana-Champaign

<sup>‡</sup> Department of Mathematics, Imperial College London, UK

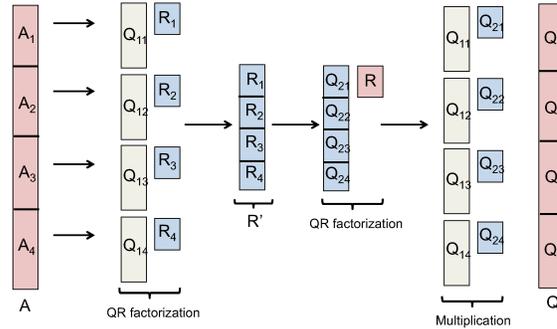


FIGURE 1: Direct TSQR algorithm suggested by Benson *et al.* (2013).

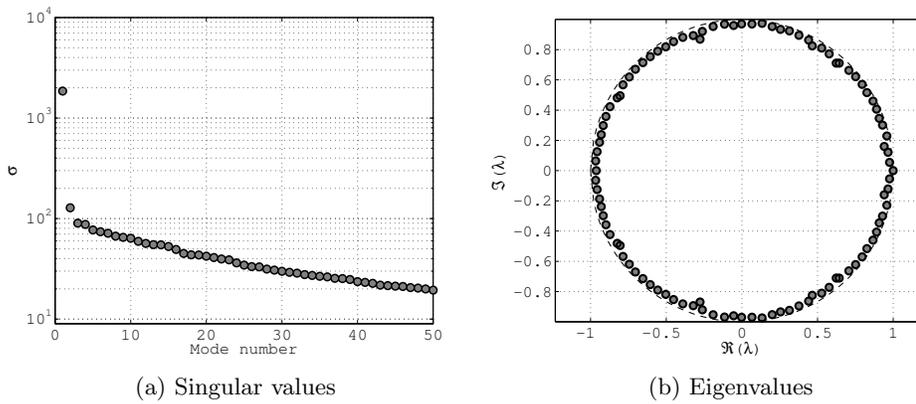


FIGURE 2: Comparison of the singular values and eigenvalues of the serial to the parallel algorithms.  $\circ$ , serial algorithm;  $\bullet$ (gray), parallel algorithm.

eventually make the SVD-based data-driven algorithm computationally expensive and ultimately prohibitive to execute on a single processor. For this reason, it is necessary to design and implement a parallel routine for performing the decomposition algorithm; this routine would eliminate the solution of an SVD for the full, large-scale data-set and would allow for the decomposition algorithm to remain computationally manageable.

In this study we apply a parallel algorithm for performing a QR-factorization, which will be used as a basis for the singular value decomposition. The algorithm will be tested on the direct numerical simulation (DNS) of H- and K-type transitions of Sayadi *et al.* (2013), the details of which are presented in Section 3.

## 2. Parallel algorithm

In this section we will first introduce the parallel QR-decomposition which is then used as a building block for all the SVD-based data-driven decomposition algorithms. Afterwards, the algorithm for finding the singular values and singular vectors is described. The resulting parallel SVD can then be used to perform POD on numerical and experimental data. Finally, the SVD is the initial step of DMD (Schmid 2010). Therefore, the additional steps of the parallel DMD will be outlined.

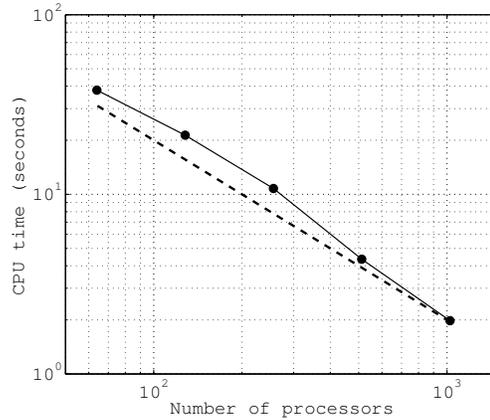


FIGURE 3: The speed-up of the parallel algorithm. — —, linear scaling; —●— scaling of the parallel algorithm.

### 2.1. Parallel QR factorization

The QR-factorization and SVD are two very fundamental decompositions that have many applications in scientific computing and data analysis. The TSQR algorithm of Demmel *et al.* (2012) is the cornerstone of our parallel DMD computer algorithm. TSQR was initially applied in the context of DMD using MapReduce programming model by Benson *et al.* (2013). The schematic of the algorithm is shown in Figure 1. Other implementations of the TSQR algorithm for the purpose of high-performance computing exist, for example Trilinos (Heroux *et al.* 2003). However, due to the simplicity of the direct TSQR method, an in-house version has been developed for a more effective integration with the underlying flow solver and further post-processing purposes. The data matrix is first divided into sub-matrices of the same column size and distributed among the processors. There are no restrictions on how to divide the data-matrix. The QR-factorization is then performed on each sub-matrix and the resulting  $\mathbf{R}_i$  matrices are gathered into a single matrix,  $\mathbf{R}'$ . A QR-factorization is then performed on the resulting  $\mathbf{R}'$  matrix. The resulting  $\mathbf{Q}$  is then divided between the processors,  $\mathbf{Q}_{2i}$  with  $i = 1, \dots, m$ , where  $m$  is the number of processors.  $\mathbf{Q}_{2i}$  is a square matrix with the size of the number of snapshots. Multiplying the resulting  $\mathbf{Q}_{2i}$  by the original  $\mathbf{Q}_{1i}$ 's of the sub-matrices gives the final  $\mathbf{Q}_i$  of the full data matrix, which is already distributed among the processors.  $\mathbf{R}$  is the upper-triangular matrix resulting from the QR-factorization of  $\mathbf{R}'$ .

### 2.2. Parallel singular value decomposition

Once the QR-factorization is computed, the singular value decomposition of the data matrix can be calculated in a straightforward manner by an additional SVD on the small matrix,  $\mathbf{R}$ , giving  $(\mathbf{U}_R, \mathbf{\Sigma}, \mathbf{W}^t) = \text{SVD}(\mathbf{R})$ . The left-singular values of the full data-matrix  $\mathbf{U}_i$  on each processor will then be computed by multiplying each  $\mathbf{Q}_i$  matrix by the resulting  $\mathbf{U}_R$ . The singular values and the right-singular vectors are identical to the ones computed from  $\mathbf{R}$ .

In order to investigate whether the results of the parallel algorithm match those of the serial one, a SVD is performed on 101 snapshots taken from a subdomain of the DNS of Sayadi *et al.* (2013), with 128, 100, and 128 grid points in the streamwise, wall-normal

---

	$N_x$	$N_y$	$N_z$	$\Delta x^+$	$\Delta y_{\min}^+$	$\Delta z^+$	$Re_{\theta_{\max}}$	Number of snapshots
H-type	4096	240	512	10.3	0.41	5.4	1250	101
K-type	4096	240	512	9.7	0.39	5.5	1410	101

---

TABLE 1: Computational domain of the respective DNS simulation used for the decomposition; total degrees of freedom per snapshot  $\approx 0.5$  Billion.

---

and spanwise directions, respectively. Figure 2(a) compares the singular values obtained from the two different algorithms and the results agree to within round-off.

Figure 3 shows the manner by which this parallel algorithm scales as the number of processors increases. Each column of the snapshot matrix consists of approximately 8 million degrees of freedom. This is the maximum size of data which would fit on 64 processors (lower limit of the scaling analysis). The maximum number of processors used for this scaling analysis is 1024. This figure shows that the scaling is near perfect for this case study.

### 2.3. Dynamic mode decomposition

Once singular values of the full data-matrix are computed, the same procedure as described in Schmid (2010) can be followed. The snapshot sequence of data is given by matrix  $\mathbf{V}_1^N$  (snapshot matrix,  $\mathbf{V}_1^{N-1}$  is the same as  $\mathbf{A}$ , given in previous sections):

$$\mathbf{V}_1^N = \{v_1, v_2, \dots, v_N\}, \quad (2.1)$$

where the vector  $v_i$  is the  $i^{\text{th}}$  snapshot of the flow field containing, e.g., the velocity field. If we assume that a linear mapping  $\mathbf{M}$  connects snapshot  $v_i$  to the subsequent snapshot  $v_{i+1}$ , we have

$$\mathbf{V}_2^N = \mathbf{M}\mathbf{V}_1^{N-1}. \quad (2.2)$$

Replacing the snapshot matrix by the singular values and vectors from the parallel SVD and multiplying both sides by  $\mathbf{U}^*$  gives  $\mathbf{U}^*\mathbf{A}\mathbf{U} = \mathbf{U}^*\mathbf{V}_2^N\mathbf{W}\Sigma^{-1} \equiv \tilde{\mathbf{S}}$ . In order to compute  $\tilde{\mathbf{S}}$ ,  $\mathbf{U}^*\mathbf{V}_2^N$  is initially formed on each processor separately and the resulting small matrix is then added across processors. Let  $y_i$  and  $\mu_i$  be the  $i^{\text{th}}$  eigenvectors and eigenvalues of  $\tilde{\mathbf{S}}$ , respectively. We then have

$$\mathbf{U}^*\mathbf{A}\mathbf{U}\mathbf{Y} = \tilde{\mathbf{S}}\mathbf{Y} = \mathbf{Y}\mu, \quad (2.3)$$

and

$$\mathbf{A}\mathbf{U}\mathbf{Y} = \mathbf{U}\mathbf{Y}\mu, \quad (2.4)$$

with  $\mathbf{Y}$  containing the eigenvectors  $y_i$  and  $\mu$  containing the eigenvalues  $\mu_i$ . The dynamic modes are calculated as

$$\Phi = \mathbf{U}\mathbf{Y}. \quad (2.5)$$

The final multiplication (Eq. 2.5) can easily be performed in parallel. It can be deduced from Eqs. (2.2)-(2.5) that, once the singular values are computed, the algorithm is parallelizable in a straightforward fashion. Figure 2(b) shows the comparison of the eigenvalues evaluated through the serial and parallel algorithms. The subdomain is the same as that described in Section 2.2.

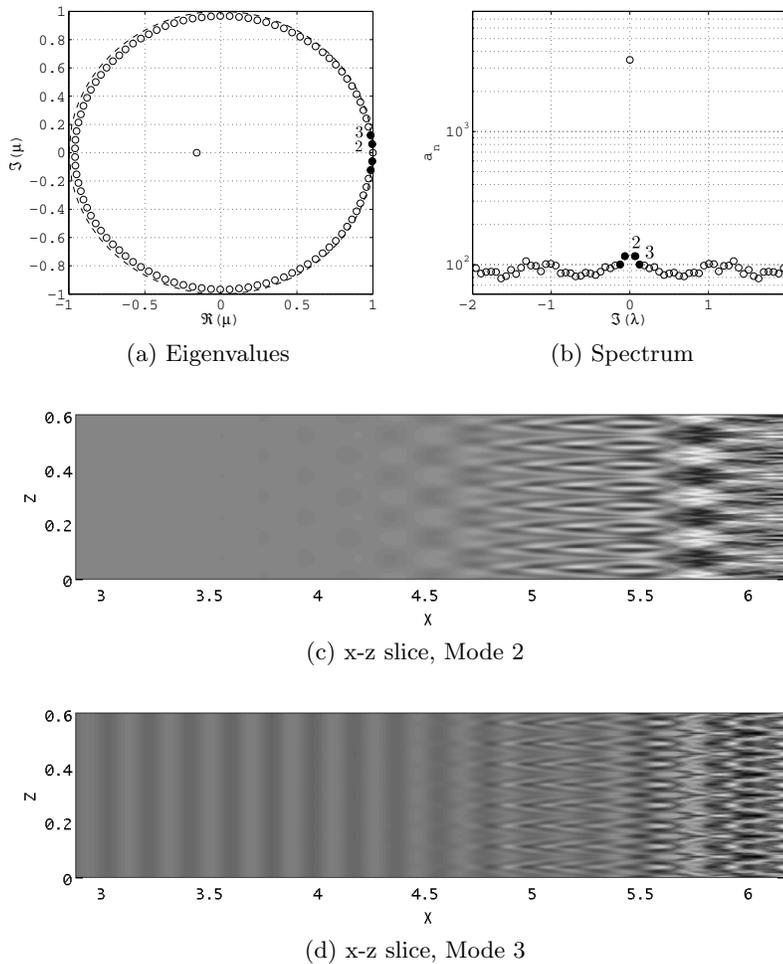


FIGURE 4: Modal decomposition of the H-type transition. (c, d) wall-parallel slices of the streamwise velocity component. The selected DMD modes correspond to the Tollmien-Schlichting and subharmonic waves.

### 3. Results

The performance of the developed algorithm will be assessed by performing DMD on the data of direct numerical simulations of H- and K-type transitions of Sayadi *et al.* (2013); details of the analyzed domains are given in Table 1. The decomposed domain consists of approximately half a billion grid points in each simulation and covers the full streamwise extent, from low-amplitude disturbances to non-linear breakdown into the turbulent regime. Previously, a domain of this size could not be analyzed due to memory restriction on a single processor.

The result of DMD of the streamwise velocity component of the H-type transition is first discussed. Hundred and one equally spaced snapshots are included in the analysis. These snapshots span approximately two periods of the fundamental Tollmien-Schlichting wave. The extracted eigenvalues are plotted in Figure 4(a). Two modes, mode 2 and mode

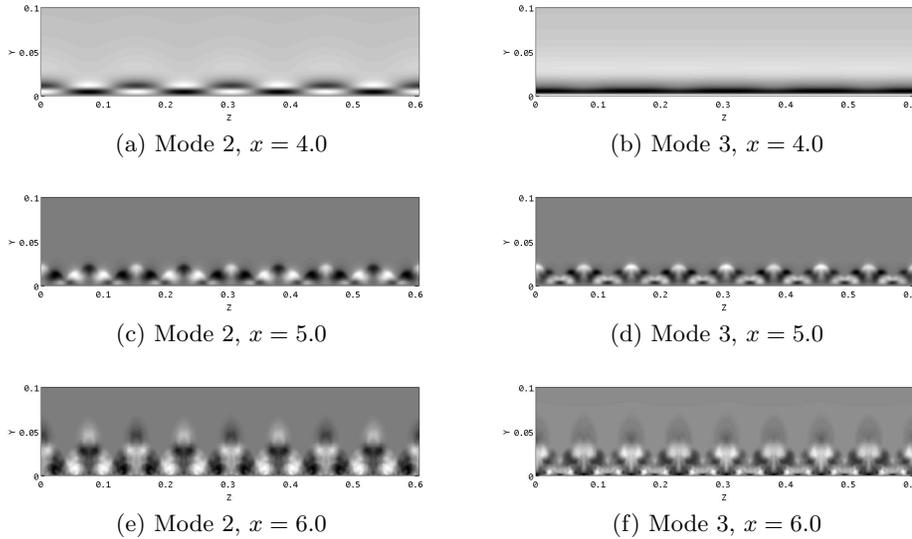


FIGURE 5: Contours of the streamwise velocity of modes 2 and 3, H-type transition.

3, are highlighted in the figure and are selected using the sparsity promoting algorithm (Jovanović *et al.* 2014) for further analysis. As demonstrated in Sayadi *et al.* (2014), these two modes coincide with the fundamental Tollmien-Schlichting and subharmonic modes of the early transitional regime. The amplitudes of the modes are determined using the optimization algorithm described in Jovanović *et al.* (2014) and the resulting spectrum is shown in Figure 4(b).

The spatial evolution of the DMD modes at a fixed wall-normal slice across the domain is given in Figure 4(c,d). The Tollmien-Schlichting wave is characterized by its two-dimensionality, which is illustrated in Figure 4(d) at the initial stage of the development of mode 2. However, as the flow evolves into the nonlinear stage of the transition mechanism, the two-dimensionality of the Tollmien-Schlichting wave breaks down. In contrast, the subharmonic wave exhibits a cosine shape in the spanwise direction at the upstream portion of the domain. The spanwise extent of the domain is such that it contains four wavelengths of the subharmonic wave, and this can be noticed in the extracted mode of Figure 4(c), especially in the streamwise region of  $4.2 \leq x \leq 4.5$ . In order to better visualize the evolution of the modes downstream, we have extracted three slices at fixed streamwise locations. The comparison of the streamwise slices of Figure 5(a,b) illustrates the three-dimensional shape of the subharmonic disturbance versus the two-dimensional nature of the Tollmien-Schlichting wave. These modes are similar to the modal shapes that are extracted through linear stability analysis at this stage of the transition process, since the amplitude of the disturbances is still relatively low. As the modes evolve downstream, mode 2 maintains the spanwise wavelength of the initial subharmonic wave. This is illustrated by the shape of this mode shown in Figure 5(c) and the alternating signs along the spanwise direction. Mode 3 evolves in a different manner, and no alternating signs are observed in the shape of the mode. Similarly, at the streamwise location of  $x = 6.0$ , close to the breakdown to turbulence, the overall shape of the modes resembles the behavior observed upstream.

Now we will present the low-frequency modes extracted through dynamic mode de-

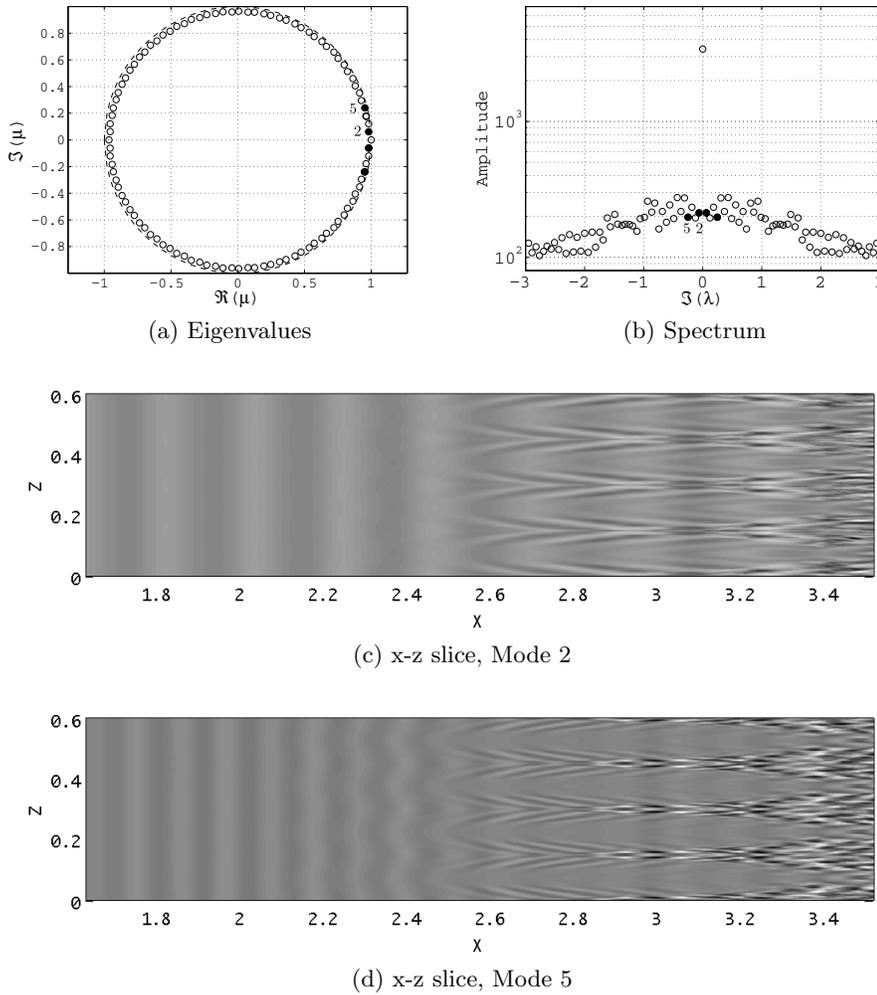


FIGURE 6: Modal decomposition of the K-type transition. (c, d) wall-parallel slices of the streamwise velocity component. The selected DMD modes correspond to the two low-frequency modes.

composition of the K-type transition data. The size of the data is similar to that of the H-type transition, described previously and given in Table 1. The eigenvalues and spectrum are computed using the sparsity promoting algorithm (Jovanović *et al.* 2014) and are plotted in Figure 6. The highlighted modes, 2 and 5, are the chosen two low-frequency modes, which will be analyzed in further detail. Mode 5 is the higher harmonic of mode 2. Figure 6(c,d) show that at the initial stage of the evolution, where the disturbances are of low-amplitude, their growth follows the linear stability approximation, both modes are two-dimensional. The lowest-frequency mode (mode 2) represents the Tollmien-Schlichting wave and has the same frequency. The higher-frequency mode appears with a smaller wavelength in the streamwise direction upstream of the domain. As these modes evolve downstream, the location where the three-dimensional trace appears

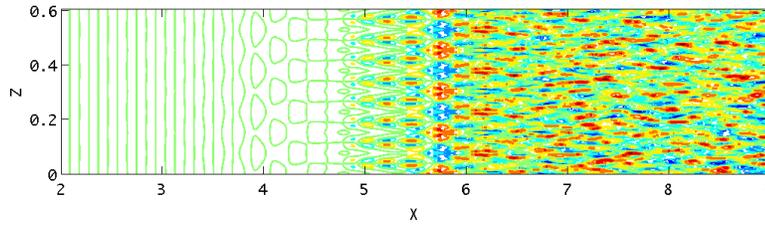
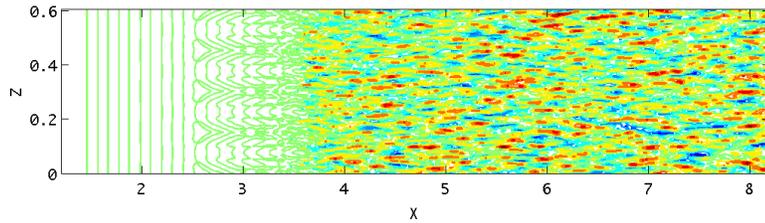
(a) H-type transition normalized by  $x_0$ , where  $Re|_{x_0} = 10^5$ .(b) K-type transition normalized by  $x_0$ , where  $Re|_{x_0} = 10^5$ .

FIGURE 7: Contours of streamwise velocity showing the development of the lowest-frequency mode (mode 2) for the H- and K-type transitions.

in both modes coincides with the location where  $\Lambda$ -vortices start to form within the computational domain. Further downstream, where the  $\Lambda$ -vortices evolve into localized turbulent patches, smaller scales also appear in the shape of these modes.

In order to examine the manner by which modes of the same frequency evolve within the two different transition scenarios, the evolution of mode 2 is compared between K- and H-type transitions in Figure 7. The wall-parallel slice includes the full computational domain, including the fully turbulent regime. The K-type transition is characterized by the appearance of aligned  $\Lambda$ -shaped vortices in the late stages of transition, whereas these vortices appear in a staggered format in the H-type transition. This difference is apparent from the shape of these modes. In addition, the trace of these  $\Lambda$ -vortices can be clearly seen in the low-frequency modes of the K-type transition. The distance from which the three-dimensional effects become visible to where the mode stops being spanwise periodic is smaller in the K-type compared to that in the H-type transition. This can be attributed mainly to the higher-amplitude initial disturbances that are introduced into the computational domain for the K-type transition, causing the region of nonlinear growth to shorten and the breakdown to turbulence to happen quicker. Although the spatial development of these modes depends on the transition scenario, in the region where the flow is fully turbulent the differences between the two modes diminish (Sayadi *et al.* 2013). In both transition scenarios the modes are composed of streaky structures in the streamwise direction. The shape and the size of these streaks is similar to what is reported for streaks in the buffer layer of turbulent wall-bounded flows. This is to be expected as the slices are taken within the buffer layer region of the turbulent boundary layer for both cases.

#### 4. Conclusions

Data-driven decompositions are efficient techniques for extracting coherent structures from data taken from experiments or numerical simulations. Most of these decomposition algorithms are based on the QR-factorization of the constructed snapshot matrix. The columns of the snapshot matrix represents the full data at a single time, and the rows are equally spaced time snapshots of the same data format. As the size of the simulations increases, the rows of the snapshot matrix also increase, resulting in a tall-and-skinny (TS) matrix. In this study we have adopted a parallel QR algorithm, proposed originally Demmel *et al.* (2012) and later used by Benson *et al.* (2013) for a MapReduce infrastructure. Once the QR algorithm is in place, the singular values and vectors can be computed in a straightforward fashion. This algorithm is shown to scale linearly on up to 1024 processors. With access to the singular eigenvalues and vectors, we can extend the algorithm to perform DMD within the same context. We have applied this parallel algorithm to the DNS data of H-type and K-type transitions, where each column of the data matrix is composed of half a billion degrees of freedom.

#### REFERENCES

- BENSON, A. R., GLEICH, D. F. & DEMMEL, J. 2013 Direct QR factorization for tall-and-skinny matrices in MapReduce architectures. *IEEE International Conference in Big Data*.
- DEMMEL, J., GRIGORI, L., HOEMMEN, M. & LANGOU, J. 2012 Communication-optimal parallel and sequential QR and LU factorizations. *SIAM J. Sci. Comput.* **34**, 206–239.
- GRILLI, M., SCHMID, P. J., HICKEL, S. & ADAMS, N. A. 2012 Analysis of unsteady behaviour in shockwave turbulent boundary layer interaction. *J. Fluid Mech.* **700**, 16–28.
- HEROUX, M., BARTLETT, R., HOEKSTRA, V. H. R., HU, J., KOLDA, T., LEHOUCQ, R., LONG, K., PAWLOWSKI, R., PHIPPS, E., SALINGER, A., THORNQUIST, H., TUMINARO, R., WILLENBRING, J. & WILLIAMS, A. 2003 *An overview of trilinos*. Tech. Rep. SAND2003-2927. Sandia National Laboratories.
- JOVANOVIĆ, M. R., SCHMID, P. J. & NICHOLS, J. W. 2014 Sparsity-promoting dynamic mode decomposition. *Phys. Fluids* **26** (2), 024103 (22 pages).
- MEZIC, I. 2013 Analysis of fluid flows via spectral properties of the Koopman operator. *Annu. Rev. Fluid Mech.* **45**, 357–378.
- ROWLEY, C. W. 2005 Model reduction for fluids using balanced proper orthogonal decomposition. *Int. J. Bifurcation and Chaos* **15** (3), 9971013.
- ROWLEY, C. W., MEZIC, I., BAGHERI, S., SCHLATTER, P. & HENNINGSON, D. S. 2009 Spectral analysis of nonlinear flows. *J. Fluid Mech.* **641**, 115–127.
- SAYADI, T., HAMMAN, C. W. & MOIN, P. 2013 Direct simulation of complete H-type and K-type transitions with implications for the structure of turbulent boundary layers. *J. Fluid Mech.* **724**, 480–509.
- SAYADI, T., SCHMID, P. J., NICHOLS, J. W. & MOIN, P. 2014 Reduced-order representation of near-wall structures in the late transitional boundary layer. *J. Fluid Mech.* **748**, 278–301.
- SCHMID, P. J. 2010 Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28.