INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **58**:1379–1405
Published online 14 April 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1792

A vortex axis and vortex core border grid adaptation algorithm

Markus Rütten[1,*,†], Thomas Alrutz[1] and Holger Wendland[2]

[1] *German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, Bunsenstrasse 10, D-37073 Göttingen, Germany*
[2] *Department of Mathematics, University of Sussex, Brighton BN1 9RF, U.K.*

SUMMARY

In the design process of hydrodynamical and aerodynamical technical applications, the numerical simulation of massively separated vortical flow is crucial for predicting, for example, lift or drag. To obtain reliable numerical results, it is mandatory to accurately predict the physical behavior of vortices. Thus, the dominant vortical flow structures have to be resolved in detail, which requires a local grid refinement and certain adaptation techniques. In this paper, a vortex flow structure adaptation algorithm is presented, which is particularly designed for local grid refinement at vortex axes positions and associated vortex core border locations. To this end, a fast and efficient vortex axis detection scheme is introduced and the algorithm for the vortex core border determination is explained. As the interaction between vortices makes the assignment of grid points to a certain vortex axis difficult, a helicity-based vortex distinction approach in combination with a geometrical rotational sensor is developed. After describing the combined different techniques in detail, the vortex feature adaptation algorithm is applied to analytical and more realistic examples, which show that the described grid adaptation algorithm is able to enhance the grid cell resolution locally such that all significant vortical flow phenomena are resolved. Copyright © 2008 John Wiley & Sons, Ltd.

Received 28 November 2006; Revised 24 January 2008; Accepted 28 January 2008

KEY WORDS: grid adaptation; CFD; vortical flow

1. INTRODUCTION

In modern aircraft and vehicle design, as well as in the development process of hydrodynamic components of technical applications, the numerical flow simulation plays an increasingly important role. In particular, one crucial point in the simulation process is the prediction of required design criteria, which have to be satisfied by an optimized solution. Moreover, the flow structure, which has to be resolved accurately, determines the turnaround times in design and optimization processes.

[*] Correspondence to: Markus Rütten, German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, Bunsenstrasse 10, D-37073 Göttingen, Germany.
[†] E-mail: markus.ruetten@dlr.de

Often, massively separated flow occurs and, as a result, vortices start to develop, which then dominate the global flow structure. These vortices have significant influence on lift, drag, or pressure loss. Once they have developed, the time-dependent evolution of the vortices determines the overall aerodynamical or hydrodynamical characteristics of the application and thus the efficiency. Hence, for the numerical simulation of flows it is essential to resolve all occurring vortices with all their main features such as the vortex axis, the core width, the vortex hull, their decay, and also the interaction between vortices.

From the physical point of view, vortices are characterized by strong gradients in pressure, velocity, and the associated flow field. This requires the numerical simulation to resolve such gradients, which is only possible with an appropriate grid cell resolution. Only if the spatial resolution is sufficiently fine to resolve the involved gradients adequately, a numerical simulation will be able to accurately predict the design objective criteria. However, in the light of computing power and economic costs it is, even nowadays, not possible to discretize the entire simulation domain uniformly at such a fine level such that all flow features are automatically detected and well represented. Hence, to achieve the overall goal, namely the improvement of the results of the numerical simulation, local grid adaptation techniques, based on physical features, as well as grid refinement, and de-refinement strategies are crucial and have to be utilized.

Unfortunately, the precise definition of the flow features themselves is still an open field of research. This makes the design of a feature-based adaptation algorithm particularly difficult. A first step in this direction has been made in [1], where the vortex breakdown over a pitching delta wing has been investigated. However, here we are more interested in the actual feature extraction.

Thus, we will concentrate on resolving two of the major features of a vortex, the vortex axis and the border of the vortex core. To this end, we present an advanced vortex core detection and analysis method combined with an adaptation technique for local grid refinement.

The paper is organized as follows. In Section 2, we will give a short and comprehensive overview of the most significant physical aspects of vortices, including methods to identify them and to localize their axis and core border. After that, we describe possible ways of how to classify and distinguish vortices. This will be useful to understand the physical properties of interacting vortices, which is necessary for the adaptation sensor used in our refinement algorithm, which is described in Section 3. The main part of this paper consists of describing the new vortex feature adaptation technique in detail (Section 4) and its application to academic test cases and to a more realistic technical configuration (Section 5). We conclude the paper by resuming the main results and indicating possible future work.

2. VORTICES

Although most scientists working in fluid dynamics have a common idea of the term *vortex*, a precise mathematical description is difficult and not complete in the sense of covering all features of this kind of rotational fluid. This lack of a general vortex definition has led to various vortex identification schemes, which are relatively successful in practice but which are mainly restricted to specific applications. Hence, we briefly discuss the vortex-related terms, pointing out that different vortex definitions and detection methods often lead to similar or identical terms. In the first part, without the intention of completeness, we give only those vortex definitions and identification schemes, which are relevant for our adaptation algorithm. Hence, we concentrate on

DOI: 10.1002/fld

purely kinematical vortex definitions, as we intend to use only the velocity field and its derivatives. In the second step, we are going to refer to those vortex axis definitions and axis detection schemes which are needed for the adaptation algorithm itself. Finally, the vortex core will be explained which is our main adaptation objective and which plays a crucial role in the physical development of the flow structure.

2.1. *Definition and detection of vortices*

Referring to the intrinsic difficulties of a general vortex definition, Lugt [2] speaks of a classical dilemma. Nonetheless, as an orientation he proposes the following definition of a vortex:

> A vortex is the rotating motion of a magnitude of material particles around a common center.

This intuitive definition describes very well the visual observations in nature. Unfortunately, it shifts the problem of defining a vortex to the problem of defining the common center of rotation. Furthermore, this definition is not valid in general, because it is not Galilean invariant under moving reference frames. Robinson [3] extended Lugt's definition by introducing a concrete observation time and a general moving reference frame.

> A vortex exists when instantaneous streamlines mapped onto a plane normal to the vortex core exhibit a roughly circular or spiral pattern, when viewed from a reference frame moving with the center of the vortex core.

Mathematically, this implies that the vortex core and its motion have to be known in advance. But the vortex core is one of the major quantities of a vortex itself and in most technical applications its location is unknown *a priori*. Robinson's definition has another major drawback: There is no formal definition of the vortex core. It can only be defined with respect to the *vortex axis*, which is the common center of rotation.

Both definitions above are Lagrangian vortex definitions. In the Lagrangian approach the motion of particles is observed. In practice, particles have to be seeded and traced. To reduce the computational complexity, the initial positions of the particles have to be chosen as a function of the just mentioned common center. This requires again a priori knowledge about the vortical flow, which is often not given. Furthermore, Robinson's approach is also not invariant under Galilean transformations. As a remedy, Cucitore *et al.* [4] proposed a vortex definition using the relative displacement of two neighboring particles over a short time interval.

For our purposes, the use of a Lagrangian vortex definition based on particle motion is not feasible. As our algorithm is intended to be incorporated into an entire computational fluid dynamics (CFD) solving process, we can only use the grid and the flow solution at a certain, fixed time and cannot use a sequence of temporal grids and solutions. Hence, a particle tracing cannot be applied. This means that we have to concentrate on Eulerian and local vortex definitions, where local flow properties are investigated at fixed locations.

Almost all Lagrangian and Eulerian vortex definitions imply that a vortex is a regional flow pattern. This leads to the assumption that it is possible to define a bounding vortex hull surface, separating the vortical rotational flow region from the outer irrotational flow. This is an ideal, theoretical point of view and only valid for ideal fluids. Based on this assumption, Saffmann [5] found an elegant way of defining vortex filaments. Unfortunately, for real fluids, with the influence of viscosity, it is impossible to have an exact separating border, in particular if the unsteady vortex evolution process includes diffusion of vorticity coupled with strain. Nonetheless, from a practical

DOI: 10.1002/fld

point of view, vortex definitions that are providing a sharp separating border are useful for confining the region, which has to be regarded.

In the search for a pointwise, local Eulerian vortex definition Levy *et al.* [6] introduced the *normalized helicity density*, often also called *stream vorticity*

$$h_{\mathrm{n}}=\frac{\mathbf{v}\cdot\boldsymbol{\omega}}{|\mathbf{v}||\boldsymbol{\omega}|} \tag{1}$$

The normalized helicity density represents the cosine of the angle between the velocity vector $\mathbf{v}$ and the vorticity vector $\boldsymbol{\omega}$. The physical idea behind employing the normalized helicity density for the vortex detection is the following one. In a vortical flow region, the vortex lines and streamlines are winding around the vortex axis. Furthermore, they are entangled. This highly complicated relationship between vortex lines and streamlines results locally in a simple geometrical relation: Near vortex core regions, the angle between the velocity $\mathbf{v}$ and vorticity $\boldsymbol{\omega}$ is assumed to be small. Following this observation, a vortical region can be defined as a region where h_{n} differs significantly from zero. This vortex definition has the following, additional advantage: The orientation of the rotation of a vortex can be determined by the sign of the helicity density. Hence, it is possible to distinguish between counter-rotating vortices. The helicity can be used to separate primary from secondary vortices. This important property will be crucial for our adaptation algorithm. Unfortunately, this criterion is sensitive to fluctuations in the data, which may come from discretization errors or coarsely resolved flow structures. As we are aiming at data sets coming from a numerical CFD computation, this is an issue we have to take care of. This can be done by using this criterion either in combination with one of the following vortex criteria or by setting appropriate thresholds. Another disadvantage is that this criterion is again not Galilean invariant, and hence only of practical relevance if a fixed reference frame is given.

Other pointwise local vortex definitions and identification schemes are based on properties of the local velocity gradient tensor $\nabla\mathbf{v}$. For example, one important feature of a rotational fluid is the existence of *complex-valued* eigenvalues of the velocity gradient tensor. This can be utilized in another vortex definition. Dallmann [7] and Chong *et al.* [8] defined a vortex using the discriminant

$$D=27R^2+(4P^3-18PQ)R+(4Q^3-P^2Q^2) \tag{2}$$

of the velocity gradient tensor $\nabla\mathbf{v}$. A positive value of the discriminant D is equivalent to the existence of complex-valued eigenvalues and thus indicates a vortex. In (2), the discriminant is defined using the three invariants P, Q, and R of the velocity gradient tensor, which we state below using the strain tensor S and rotation tensor Ω, i.e.

$$S=\tfrac{1}{2}(\nabla\mathbf{v}+\nabla\mathbf{v}^{\mathrm{T}}),\quad \Omega=\tfrac{1}{2}(\nabla\mathbf{v}-\nabla\mathbf{v}^{\mathrm{T}})$$

and the eigenvalues $\lambda_1,\lambda_2,\lambda_3$ of $\nabla\mathbf{v}$. The first invariant P is given by

$$P=-\mathrm{trace}(\nabla\mathbf{v})=-\sum_{i=1}^{3}S_{ii}=-(\lambda_1+\lambda_2+\lambda_3) \tag{3}$$

For the velocity field $\mathbf{v}$, this corresponds to the negative sum of the principle rates of strain or the negative divergence of the velocity field, respectively. For an incompressible flow P vanishes.

DOI: 10.1002/fld

The second invariant Q is the sum of the product of the eigenvalues of the sub-determinants of $\nabla\mathbf{v}$ and compares the influence of pure shear to the influence of fluid element rotation

$$\begin{aligned} Q &= \frac{1}{2}[P^2 - \mathrm{trace}(\nabla\mathbf{v})^2] \\ &= \frac{1}{2}\left[P^2 - \sum_{i,j=1}^{3}(S_{ij}S_{ji} + \Omega_{ij}\Omega_{ji})\right] \\ &= \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1 \end{aligned} \tag{4}$$

For completeness, we also state the third invariant R, which describes the stability of the local flow structure. It is given by the negative determinant of the velocity gradient tensor, i.e.

$$\begin{aligned} R &= -\det(\nabla\mathbf{v}) \\ &= \frac{1}{3}[-P^3 + 3PQ - \mathrm{trace}(\nabla\mathbf{v})^3] \\ &= \frac{1}{3}\left[-P^3 + 3PQ - \sum_{i,j,k=1}^{3}(S_{ij}S_{jk}S_{ki} + 3\Omega_{ij}\Omega_{jk}\Omega_{ki})\right] \\ &= -\lambda_1\lambda_2\lambda_3 \end{aligned}$$

Dallmann [9] and Hunt *et al.* [10] suggested a vortex definition only based upon the second invariant Q of the velocity gradient tensor. An invariant $Q>0$ means that rotation dominates shearing and indicates a vortex. This physical interpretation becomes more obvious when considering the kinematic vorticity number N_k introduced by Truesdell [11]. The kinematic vorticity number is defined as the dimensionless scalar

$$N_\mathrm{k} = \frac{\|\Omega\|}{\|S\|} = \frac{1}{\sqrt{2}}\frac{|\boldsymbol{\omega}|}{\|S\|} \tag{5}$$

using the Frobenius norm $\|\Omega\|$ of Ω, i.e.

$$\|\Omega\|^2 = \mathrm{trace}(\Omega\Omega^\mathrm{T}) = \sum_{i,j=1}^{3}\Omega_{ij}\Omega_{ij}$$

and the relation $\|\Omega\|^2 = \frac{1}{2}|\boldsymbol{\omega}|^2$.

A point belongs to a vortex region if the rotational part of $\nabla\mathbf{v}$ out-balances the shear part and this is given by $N_k>1$. For incompressible flow both definitions are equivalent expressions. Using the symmetry of S and the anti-symmetry of $\Omega = -\Omega^\mathrm{T}$, we see that (4) can be rephrased as

$$2Q = -\|S\|^2 + \|\Omega\|^2 \tag{6}$$

which leads to

$$1 + \frac{2Q}{\|S\|^2} = \frac{\|\Omega\|^2}{\|S\|^2} = N_k^2 \tag{7}$$

Obviously, for an invariant Q larger than zero the kinematic vorticity number will be larger than one and *vice versa*.

DOI: 10.1002/fld

A more sophisticated vortex definition comes from Jeong and Hussain [12], who introduced the λ_2-criterion. After a compelling derivation they formulated the following condition for a vortex-related pressure field:

$$\Omega^2+S^2=-\frac{1}{\rho}(\partial_{ij}p) \tag{8}$$

Following Jeong and Hussain [12], a vortex exists at locations, where the symmetric rotational vortical flow and the related symmetric shear produces a pressure minimum. However, they also point out that this vortex-induced pressure minimum may not be identical with the original pressure minimum in the flow domain. The latter one can also be caused by other superposed effects such as shocks, strong shear, or acceleration. The symmetric tensor Ω^2+S^2 has three real eigenvalues $\lambda_1\geqslant\lambda_2\geqslant\lambda_3$. To satisfy the condition of a pressure minimum, two eigenvalues have to be less than zero. This can be used as a formal definition for the vortex.

A vortex is a region where $S^2+\Omega^2$ has two negative eigenvalues, i.e. where $\lambda_2<0$.

In a recent discussion about local vortex definitions between Wu *et al.* [13] and Chakraborty *et al.* [14], the latter one derived the so-called *inverse spiraling compactness* criterion, which is based on the observation that a vortical flow region may be characterized by the ratio of the real part to the imaginary part of complex eigenvalues. This criterion is a measure for the spatial extent of the local spiraling motion. In particular, Chakraborty compared his criterion with the vortex criteria and definitions above with respect to their regional extend and he showed that, although all criteria predict slightly different bounding vortex hull surfaces, they all cover the main features: the vortex core border and the vortex axis. He also noted that his *inverse spiraling compactness* criterion gives the largest region in contrast to the Q criterion, which predicts the smallest vortical-related flow region. In our case, the exclusion of borderline cases is an important advantage, which can improve the distinction between different vortices. In fact, for our purposes, the investigation of a smaller region has the additional advantage of reducing the computational complexity.

For our goal of detecting the vortex axis and vortex core border, the discussion above leads to the following two conclusions. First, we can use the normalized helicity density to distinguish between different vortices. Second, we can, in principle, use one of the local vortex definitions to restrict the flow domain, to which we have to apply our adaptation algorithm. The last point has a significant impact on the number of points we have to consider for our algorithm. Finally, although the λ_2-criterion predicts nearly the same vortex region, it involves the computation of eigenvalues. The Q criterion can be computed faster and is thus our preferred choice.

Finally, it is worth mentioning that besides the vivid discussion on a *Galilean* vortex definition, there is also an ongoing debate on a possible *objective* vortex definition, see, for example, the work of Haller [15]. However, all existing objective vortex criteria are based on higher-order derivatives. In our situation, where we want to improve the grid resolution especially at locations, where the grid is rough, a calculation of higher-order derivatives is problematic because of the lack of accuracy. Unfortunately, this automatically excludes more advanced vortex criteria.

2.2. *The vortex axis and its detection*

The starting point for our vortex feature-based adaptation algorithm will be the detection of the *vortex axis*. Unfortunately, as in the case of the vortex, there is no exact definition for the vortex axis. Banks and Singer [16] numerically integrated vortex lines, where, in addition, each newly calculated position is corrected according to a sectional pressure minimum. Miura and Kida [17] presented a similar approach based on a minimum of a reduced pressure field. Kenwright and Haimes [18] concentrated on lines following the sectional maximum of vorticity magnitude. Sujudi and Haimes [19] developed a technique, where the vortex axis line is identified with locations, where the velocity vector points in the same direction as the real eigenvector of the velocity gradient tensor. They implicitly assumed that the other two eigenvalues are complex conjugated and that in the region of complex eigenvalues a fluid element rotates around the real eigenvector. Roth [20] and Peikert and Roth [21] generalized this concept and developed a so-called *parallel vectors operator*, which analyzes the geometrical relations between two appropriate vector fields on grid cell faces to identify extremal lines, ridge and valley lines as well as the vortex axis. In particular, this parallel operator yields positions on the faces where the associated vectors are parallel.

Queued cell face points, where the parallel condition is satisfied, are connected to the vortex axis. However, their algorithm is sensitive to noisy data and often advanced filter techniques have to be applied to build the vortex line. Another advanced approach comes from Sahner *et al.* [22], who introduced a so-called *feature flow field* based on the λ_2 vortex definition. A detailed description can be found in [23]. Although they have a very convincing concept, their method has the drawback that second-order derivatives have to be computed.

However, in our application, the detection of the vortex axis is the most important step. All the following steps will be based on it. As we might also have a coarse grid resolution, i.e. a rather bad resolution or even noisy data, we cannot use any method employing higher-order derivatives. We also wish to avoid filtering. Thus, we will apply a combinatorial vortex axis detection algorithm, which was presented in [24]. The basic idea of this algorithm can be described as follows. We introduce a cut plane and search for points on that plane having a velocity vector parallel to an associated derivative vector of the velocity. This condition defines the location where the vortex axis crosses the cut plane. The determined points on the plane are used as starting points for a streamline integration into the whole 3D simulation domain, assuming implicitly that a vortex axis is also a streamline.

Compared with the algorithm in [24], one important improvement of our vortex axis calculation algorithm is the following one. Whenever the integration process crosses a volume cell face, we verify whether both vectors are parallel. This guarantees, on the one hand, that we are further integrating the vortex axis. On the other hand, we minimize the absolute error of vortex axis line integration by avoiding accumulation of integration errors. Moreover, we can use this check for parallelism as a termination criterion for the integration process. If the parallel check sequentially fails at more than a certain number of crossed cell faces then we will stop integrating. Another verification is done by repeating the complete vortex axis calculation procedure for a cut plane, which is slightly shifted in space. This allows us to detect vortex axes, which could not be found before. If a newly integrated vortex line hits the same grid cells as a formerly detected axis then this axis will not be considered. Consequently, this approach leads to the calculation of vortex axes for the main vortices, i.e. the primary, secondary, and tertiary vortices [24]. In this paper, we have tested the improved vortex axis algorithm in two cases, using two different derivative vectors of the velocity vector for our test for parallelism. In the first case we compared the velocity vector to the acceleration vector. In the second case we compared the velocity vector to the vorticity vector. Both cases lead to nearly the same axes. However, as our vortex distinction criterion is based on the helicity density we will use the vorticity vector as input for the parallel operator.

DOI: 10.1002/fld

2.3. The vortex core

As mentioned above, we understand a vortex to be a region with a sharp bounding hull surface, establishing a separating border between the region with a vortex and the region without a vortex. A *vortex core* is also a region. This region comprises the inner part of the vortex from the vortex axis to locations, where locally the magnitude of circumferential velocity reaches its maximum. In the outer part of the vortex, outside of the vortex core region but within the vortex hull surface, the circumferential velocity decreases again. A formal definition is given by Lugt [25], who considered solutions of the axial symmetric Navier–Stokes equations. From the analytical point of view, the maximum of the circumferential velocity magnitude denotes the vortex core border. Hence, this local maximum can be used as a suitable physical parameter to determine a hull surface comprising the whole vortex core.

Within the vortex core, the impact of dynamical forces results into a low pressure region with a minimum almost at the vortex axis. This is a result of the cyclostrophic equilibrium between the centripetal force coming from the acceleration of the fluid elements and the radial pressure gradient. Within the core region, directly at the vortex axis, the fluid rotates like a rigid body. Closer to the vortex core border, the influence of the viscosity and thus the shear part changes. Outside the vortex core but within the vortex hull, the circumferential velocity reaches an inflection point, where the vortex starts to change its basic flow behavior again. The influence of vortical motion-related shear vanishes and the vortex becomes more and more a circumferential velocity distribution such as a potential vortex. Another physical aspect can be investigated by observing the temporal or spatial development of the vortex core radius. When the core radius grows, the vortex decays and vorticity is transported to the outer direction resulting also into a decreasing radial pressure gradient. As a result, this leads to an increasing axial pressure gradient and a decreasing axial velocity which leads to a wake-like axial velocity component. Later on, the vortex

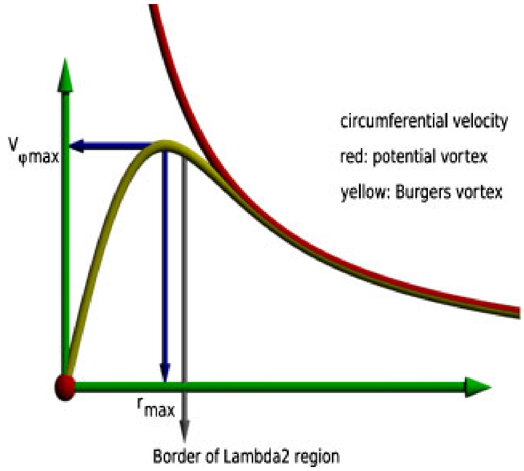

Figure 1. Sketch of the core radius and the circumferential velocity component of a Burgers vortex. The core radius is determined by the maximum of the circumferential velocity ($r_{\max}$), whereas the inflection point of the velocity is the location where the Burgers vortex becomes a potential vortex [26].

DOI: 10.1002/fld

breakdown phenomenon may occur. In the other case if the vortex core radius decreases, the vortex tube will become constricted. Then, vorticity is transported into the direction of the vortex axis, and, eventually, the vortex will become stronger with a higher circumferential velocity component. Because of this, the radial pressure gradient increases as well as the axial velocity. This finally leads to a jet-like axial velocity profile.

The physical significance of the vortex core border is reflected in the fast change of the derivatives of the circumferential velocity component. If we exemplarily look at an analytically defined vortex such as the Burgers vortex, which is a solution to the axis-symmetric Navier–Stokes equations, we may obtain an impression of the specific requirements for a grid refinement. In Figure 1, the curvature progression of the circumferential velocity component illustrates the strong flow structure change at the vortex core border.

This shows that an *a priori* globally refined grid topology for the numerical simulation is in general not possible. Thus, a dynamic grid adaptation is required.

From the discussion above, we can identify further significant difficulties we have to face. Along a developing vortex, different local circumferential velocity magnitudes will occur, as well as different vortex core radii from the current axis point position to the vortex core border. This has to to be taken into account when formulating an appropriate adaptation sensor.

3. GRID ADAPTATION ALGORITHM

We are now explaining in detail which adaptation technique we will use for refining and de-refining the grid cells. After that we will formulate the vortex analysis algorithm and the sensor for our grid adaptation.

We will employ the adaptation module of the DLR TAU-Code. This module comprises three different components for various grid manipulations, which can be used to adapt a given grid to the solved flow field (see [27, 28]):

- y_+-based grid adaptation for adjusting the first wall distance over turbulent surfaces in hybrid grids;
- hierarchical grid refinement and de-refinement for introducing new grid points using a given edge-indicator function without producing hanging nodes;
- surface approximation and reconstruction for curved surfaces after the introduction of new grid points.

For the refinement of vortices the y_+-based grid adaptation and the surface approximation are of minor interest.

Hence, we will concentrate on using the grid refinement and de-refinement algorithm together with the edge-indicator function for detecting regions, which require local refinement or de-refinement.

3.1. *The refinement and de-refinement algorithm*

A local refinement strategy has to specify how to detect grid areas, which need refinement, and a method of element subdivisions, which are a consequence of the insertion of new points into these areas.

DOI: 10.1002/fld

A very rough description of the basic refinement algorithm is given by

1. Build the edge list and the element to edge references.
2. Evaluate the edge-indicator I_e for every edge.
3. Refine the edge list, taking the edge indicators, the target point number and the grid conformity into account.
4. Calculate the coordinates of new points.
5. Construct new elements.
6. Interpolate the solution to the new points.

The refinement/de-refinement algorithm is based on an edge-based data structure. This means that the refinement indicators are evaluated for all edges. New points are chosen as the edges' midpoints; the element subdivisions are determined by the configurations of the refined edges. Step 3 of the algorithm consists of two nested loops. The outer refinement loop finds the limit L for the indicator which results in the target number of new points if all edges with $I_e > L$ are initially marked for refinement. The inner loop ensures consistency by checking all elements for a valid subdivision and by inserting additional points by marking the corresponding edges in the case of a violation. This loop has to be repeated until no more edges are marked. To prevent the propagation of the local refinement through the grid all possible refinement cases for tetrahedra are implemented as well as two anisotropic directional refinement cases for prisms. For pyramids there is only a small subset of possible refinement cases implemented because of grid consistency reasons. For a detailed description of all available refinement cases, see [27–29]. After the determination of the final edge subdivision all marked edges are identified with new point numbers and the coordinates of these new points are calculated (step 4). The new elements are constructed using the appropriate subdivisions in step 5. Finally, the solution is interpolated to the new points (step 6). For the de-refinement, the algorithm keeps track of the complete refinement hierarchy for each element in the grid. For our edge-based element representation this means that all parents, which are one level above the actual grid elements, have to be considered for the calculation of the global edge list in order to allow the edge-indicator function to select these edges for de-refinement. The preliminary step 1 of the basic algorithm is extended to the following algorithm:

1.1 Determine the direct parents of the actual elements.
1.2 Build the edge list of elements and their direct parents. Build the element to edge reference of elements/parents.
1.3 Restore midpoints of the parent edges.
1.4 Flag out all edges that are not part of an isotropic refinement.
1.5 Flag out all edges that have a parent point for de-refinement.

Step 1.1 is easily done by flagging all parents that are referenced to by other parents and then considering only the remaining parents. Step 1.2 is basically the former step 1. Steps 1.3–1.5 ensure consistency of the basic algorithm. Finally, step 3 has to be modified to support de-refinement:

3.1 Set an initial guess for the indicator limit L.
3.2 Mark all parent edges with $I_e < L$ for de-refinement.
3.3 Mark all edges with $I_e > L$ for refinement.
3.4 Mark additional edges for refinement to ensure valid refinement cases (consistency loop).
3.5 Count the marked edges. If the number of new points is too large or too small modify L, reset all marked edges and go to 3.2.

DOI: 10.1002/fld

To avoid the degeneration of elements, non-isotropically refined elements are not further refined. If such an element has to be refined, because some of its edges are marked for refinement, the direct parent element is isotropically refined and the remaining edge marks are considered for the resulting children. This guarantees stable sequences of element refinements.

3.2. The edge-indicator sensor functions

For local grid refinement/de-refinement a proper sensor or indicator is necessary. There are several approaches to define such a sensor. One of the most common approaches is to use a residual-based indicator [30]. There are also approaches to use an adjoint sensor or to use gradients or differences of an appropriate flow variable. In the implementation of the adaption module of the TAU-code, there are edge-indicator functions based on the gradient or various difference types. These functions are defined as follows.

The approximate gradient $G\Phi$ of a function Φ along an edge e with vertices p_1 and p_2 is given by $G\Phi(e)=\Delta\Phi(e)/h$. Here, $\Delta\Phi(e)$ is the difference between the point values of Φ at the two vertices p_1 and p_2 of the edge e, i.e. $\Delta\Phi(e)=\Phi(p_1)-\Phi(p_2)$, and $h=|p_1-p_2|$ is the length of the edge e. Then, one way of defining an indicator function is given by setting

$$I_{\mathrm{e}}=\Delta\Phi h^{\alpha}=G\Phi h^{\alpha+1} \tag{9}$$

with a parameter $\alpha>0$ at our disposal. This parameter is often simply set to $\alpha=1$ such that $I=G\Phi h^2$. An advantage of such a scaling parameter α is that, for a proper choice, the refinement will automatically stop after several iterations in the designated area. For $N_\phi+1$ different flow variables, we choose the following modification of $\Delta\Phi$ for the indicator function along an edge e of the grid:

$$\Delta\Phi(e)=\max_{0\leqslant i\leqslant N_\phi}\left(c_{\phi_i}\frac{\Delta\phi_i(e)}{(\Delta\phi_i)_{\max}}\right) \tag{10}$$

The weights c_{ϕ_i} are scaling parameters that give a sufficient flexibility to choose different combinations of the single flow variables of the indicator (for example, setting one c_{ϕ_i} to zero simply corresponds to 'switching' ϕ_i off).

The reference values $(\Delta\phi_i)_{\max}$ are given by

$$(\Delta\phi_i)_{\max}=\max_{0\leqslant j\leqslant N_{\mathrm{e}}}\Delta\phi_i(e_j)$$

where $N_{\mathrm{e}}+1$ denotes the total number of edges in the grid and $e_0,\ldots,e_{N_{\mathrm{e}}}$ is an enumeration of all these edges. The maximum values guarantee a balanced scaling of each contribution to the indicator function. As the edge-indicator function we can use, for example, one of the following differences:

1. The differences (Δ_{d}) of the flow values

$$\Delta_{\mathrm{d}}\phi_i(e)=|\phi_i(p_1)-\phi_i(p_2)|$$

2. The differences of the gradients (Δ_{g}) of the flow values

$$\Delta_{\mathrm{g}}\phi_i(e)=|\nabla\phi_i(p_1)-\nabla\phi_i(p_2)|$$

3. The differences of the reconstructed flow values (Δ_r) to the edge midpoints
$$\Delta_r \phi_i(e) = |(\phi_i(p_1) + \tfrac{1}{2} x_e \cdot \nabla \phi_i(p_1)) - (\phi_i(p_2) - \tfrac{1}{2} x_e \cdot \nabla \phi_i(p_2))|$$
where $x_\mathrm{e} = p_1 - p_2$.

Any of the flow variables of the solver output can be used with these edge indicators. This means all primitive and all additionally user-defined flow variables can be employed (e.g. x-, y-, z-velocity, total pressure, vorticity, pressure coefficient c_p, etc.).

Unfortunately, for the refinement of vortex cores and vortex borders this technique is insufficient. In [1], we have shown how to use the above-described sensor function with the total pressure p_tot as the flow variable to automatically refine a given grid in regions of total pressure losses. With this method we were able to resolve all significant flow features, but it was necessary to adjust additional parameters to steer the refinement process. Furthermore, that method is less precise than the one presented in this paper.

In order to employ the sensor described in Section 4 with the adaptation tool, a new indicator function has to be developed. The edge indicator defined in (10) will be redefined as
$$\Delta\Phi(e) = \max(\phi(p_1), \phi(p_2)) \tag{11}$$

4. THE VORTEX ANALYSIS AND THE ADAPTATION SENSOR

In the sequel, we will explain the main part of our vortex axis and core border adaptation algorithm. Having in particular hydrodynamical and aerodynamical applications in mind, we will, for simplicity, restrict ourselves to columnar vortices. This means particularly that we can assign a rotational sense to each vortex.

As lined out in the last section, we have to construct an adaptation sensor that delivers signals for refining those grid cells, where the numerical flow solution indicates a vortex axis or the vortex core border.

The main idea of our adaption algorithm is to assign a value between 0 and 1 to each vertex of the given grid. Cells having a vertex with a value larger than a given threshold will then be considered for refinement.

However, if we consider the typical graph of the circumferential velocity component of the Burgers vortex as a function of the radius, then we see that the radial velocity derivative becomes zero at the vortex core border, because the maximum of the circumferential velocity is attained here. Before and behind the core border the gradients are very steep. Thus, we have to adapt those cells with small radial velocity gradients in contrast to the usual procedure to adapt cells with steep gradients. Therefore, we have to construct an algorithm, which is not based on derivatives, but also delivers signal for indicating the grid cells or their edges.

The main steps of our algorithm can be summarized as follows.

1. To enhance efficiency, a filtering step based on the Q-criterion ($Q>0$) is employed to identify points that are located within a vortex. Only these points are considered in the following steps. The other points and thus their corresponding grid cells will be marked as not adaptable.
2. The vortex axes locations are computed. This is done by the following sub-steps.

   (a) First, a cut plane is calculated. Its base point can, for example, be determined by finding a global minimum of the λ_2 values or by finding the global maximum of the Q values. The

DOI: 10.1002/fld

plane's normal may be defined by the free stream velocity direction. This gives reasonable results especially for delta wing applications. In some cases it might be necessary to set more than one cut plane depending on the configuration and the flow separation. By cutting the elements of the tetrahedralized grid we finally obtain a plane consisting only of triangles.

(b) Then, we map the velocity and the derived vector field onto the plane. The derived vector field could be the acceleration, the real eigenvector of the velocity gradient tensor or the vorticity field. Its actual choice depends on the expected curvature of the vortex axes (see the discussion in Roth [20]). Moreover, the eigenvalues of the velocity gradient tensor are also mapped onto the plane, providing further vertex information.

(c) Next, we apply the parallel vector operator only to those triangles of the cut plane, where the vertices have a velocity gradient tensor with complex eigenvalues. In the case of existence, the parallel operator yields the positions on the triangles where both vectors are parallel. These positions are used as the starting points of a streamline integration for the vortex axes. The integration is done in both directions, upstream and downstream, such that the streamlines extend to both directions.

(d) During the integration procedure grid cells are crossed and marked as processed. Whenever a grid cell is left and a new cell is reached the integration process stops at the cell side face and the parallel operator is applied to this face; the integration process is restarted with the new initial position given by the parallel operator. This avoids the accumulation of integration errors and hence improves the accuracy of the algorithm.

(e) The integration process stops when the parallel operator does not find any further positions. However, since we handle noisy data we allow a certain number of failure signals when crossing a cell face.

(f) Depending on the noise level of the data the whole vortex axis calculation process (steps (a)–(e)) is repeated for a new cut plane slightly shifted to the original one. Only when the new vortex axis integration does not hit marked grid cells the newly integrated vortex axis is accepted.

All cells that are crossed by a vortex axis are marked for adaptation.

3. Once the vortex axes are defined, each grid point with $Q>0$ is assigned to an appropriate vortex using the normalized helicity density, proximity, and local flow direction.
4. To determine the vortex core radius, for each assigned grid point, the sensor value is computed using the local tangential velocity normalized by the local circumferential velocity for the vortex. Any values above a given threshold are set to unity.

The last two steps are explained in detail in the remainder of this section.

4.1. Rotational sense of vortices

As we have seen, the vortex core can only be detected when the associated vortex axis is already known. However, for general complex flow configurations more than one axis may and will occur. In fact, primary dominant vortices are accompanied by secondary or tertiary vortices or by other up-rolling vortical structures generated by instabilities in the flow field. In addition to that, vortex interaction is inevitably enforced by the vortex induction law. Therefore, we have to distinguish between single vortices to be able to calculate the right vortex core. As outlined in Section 2.1, the normalized helicity density criterion can be used to determine the rotational sense of the

DOI: 10.1002/fld

vortex. Positive values in a flow region indicate a clockwise rotation of the fluid elements around the vortex axis, when looked at the flow in downstream direction. Accordingly, negative values in downstream direction indicate counterclockwise rotation. This gives the two principal types of vortical structures: clockwise and counterclockwise rotating vortices. Thus, we calculate the normalized helicity density for all points in a vortical region. Additionally, we interpolate the normalized helicity density onto the vortex axis points.

4.2. Point to vortex assignment

After having determined the axes, we have to assign the single grid points to their *associated* vortex axis. This can be considered as a specific *nearest neighbor* problem. As each vortex axis is given by a set of points, we can approximately determine the distance of a point to a given vortex axes by finding its nearest neighbor from such a set.

To find the nearest neighbors, we first built a search tree structure for the points of each vortex axis separately using *kd-trees*. Such trees can be built efficiently and each nearest neighbor search can be executed efficiently, see, for example, [31]. Then, for a given grid point, the results of the searches can be compared and the axes with the least distance is a good candidate for the point to be assigned to.

As a further improvement, the vortex axis point search trees are sorted into two lists: one list for clockwise and one list for counterclockwise rotating vortices. As the normalized helicity density at the current grid point indicates its rotation orientation, we actually only have to compare with those axes, which are in flow regions with equally signed normalized helicity. This allows us to reduce the computational complexity even further.

However, though we can determine the right type of vortex axis in this way, we are now confronted with another problem. Often vortical flow structures are dominated by a strong primary vortex accompanied by sub-vortices with the same rotational sense which, as a consequence, have the same helicity density sign. An impression of such vortical structures can be gained when looking at the flow over a delta wing in the cut plane topology, which is depicted in Figure 14. Unfortunately, the sub-vortices are significantly smaller in their core radii than the primary vortex. Consequently, a grid point may have a shorter distance to a sub-vortex axis, but belongs actually to the core of the primary vortex axis. Thus, we cannot simply use a point to vortex axis point distance algorithm, as described above. We also have to take the different scales of the core radii into account.

To this end, we introduce a geometrical method to identify the correct vortex axis. For a given grid point, we first determine the *two* nearest vortex axes of equal rotation as described above. Next, we connect the grid point to each of the nearest neighbors from both vortex axes. This gives us two vectors and our criterion will be based on the angle between them. If this angle is less than 90° then we assign the grid point to the nearest vortex axis. Figure 2 shows this geometrical relation of the principal flow behavior.

Otherwise, if the grid point is located between both vortex axes and the angle is larger than 90° then we use the local rotational orientation as additional information to determine the correct vortex axis. This case is illustrated in Figure 3.

Let us consider a 2D projection of the flow between the two co-rotating vortices equipped with an appropriate coordinate system. Then, we can observe that the flow of one vortex is going upwards and, separated by a strong shear layer, the flow of the other vortex is going downwards (see Figure 4). We use this observation in our assignment algorithm.

DOI: 10.1002/fld

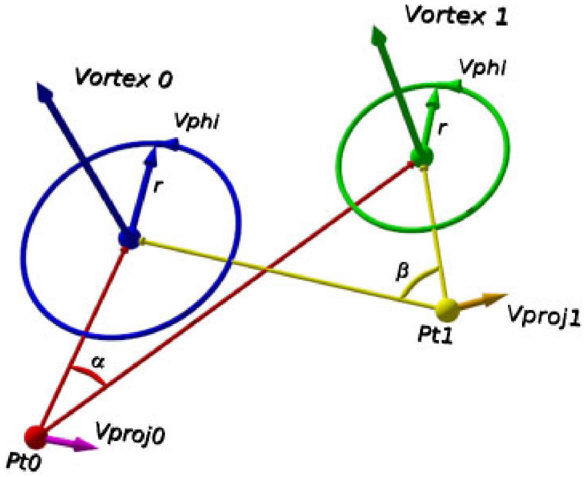

Figure 2. Sketch of the principal flow geometry between co-rotating vortices.

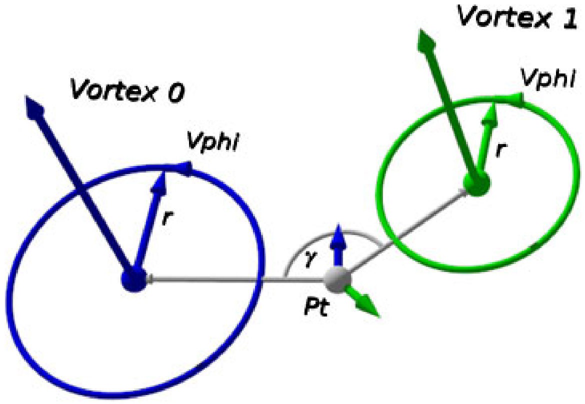

Figure 3. Critical point–vortex axis assignment case.

To this end, we introduce a local coordinate system for each vortex axis. The local radial unit vector $\mathbf{e}_\mathrm{r}$ of such a coordinate system is given by the normalized vector being orthogonal to the vortex axis and pointing from the vortex axis to the grid point. The axial unit vector $\mathbf{e}_z$ points from the vortex axis into downstream direction along the vortex axis. By calculating the cross-product of these two vectors we can define the circumferential direction vector

$$\mathbf{e}_\varphi := \frac{\mathbf{e}_z \times \mathbf{e}_\mathrm{r}}{|\mathbf{e}_z \times \mathbf{e}_\mathrm{r}|}$$

Next, for both axes, we project the velocity vector of the given grid point onto the plane spanned by $\mathbf{e}_\mathrm{r}$ and $\mathbf{e}_\varphi$ and compute the inner products between the projected velocity vector and the circumferential unit vectors $\mathbf{e}_\varphi$. Now there are two possibilities depending on the angle between $\mathbf{e}_\varphi$ and the projected vector and the rotation orientation of the vortices. First, let us consider two

DOI: 10.1002/fld

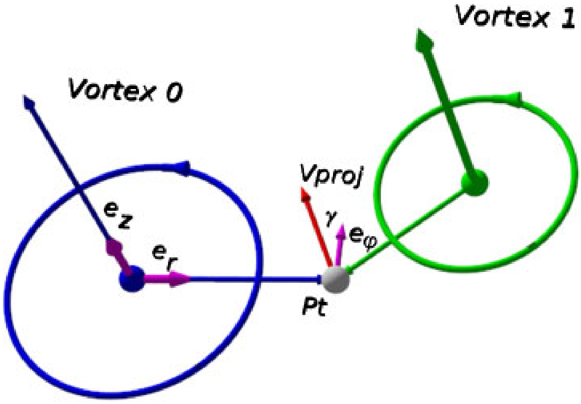

Figure 4. Sketch of the geometry of the point assignment algorithm. Point Pt is assigned to vortex 0.

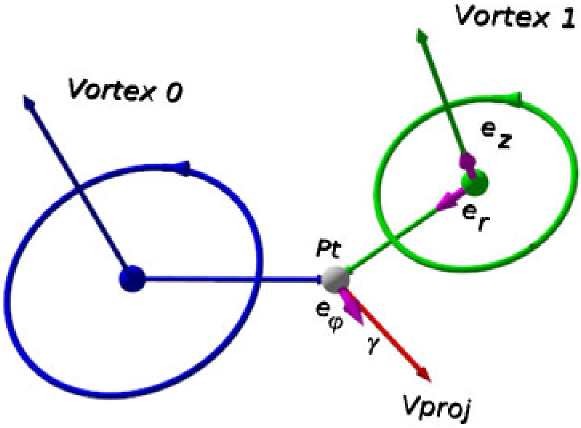

Figure 5. Sketch of the geometry of the point assignment algorithm. Point Pt is assigned to vortex 1.

vortices rotating counterclockwise in downstream direction, as shown in Figures 4 and 5. Here, the grid point belongs to that region, where the angle is less than 90° or, equivalently, the inner product is positive. If the inner product is negative, the grid point does not belong to this vortex.

In the same way, for two vortices rotating clockwise in down stream direction, the grid point belongs to that vortex region, where the angle is larger than 90°, or where the inner product is positive.

This criterion requires that both vortex axes are more or less parallel, which is the case for the dominant vortices above a delta wing. In flow fields where neighboring vortex axes are tilted we have strong shear flows in between. This flow region is excluded by the vortex criteria explained above. Therefore, the algorithm can be extended by the constraint that a point can only be assigned to an axis if all grid points between the point and axis are assigned to a vortical region by using one of the explained criteria.

In this way, we can uniquely assign each grid point to a certain vortex axis point.

During the assignment process, we count how often each vortex axis point is employed. This additional information will later on be used for the normalization of the circumferential velocity components.

4.3. The vortex core border detection

With assigning each grid point to its vortex axis point, we have also for each point in a vortical region the appropriate local vortical coordinate system. This is used to determine the circumferential component of the velocity vector at each grid point. For constructing our unique sensor value needed by our adaptation algorithm, we have to normalize these velocity components. In order to obtain a normalization factor we construct for each grid point a ray pointing from the associated vortex axis point over the grid point itself outwards in radial direction. On this ray we calculate the maximum circumferential velocity component going stepwise from the axis outwards to the border of the vortical region. The step size is determined by a fifth of the local grid cell size. Now we can normalize the circumferential velocity components of all grid points by the calculated maximum. After all these steps we obtain the adaptation sensor field consisting of values from 0 as lower limit to values close to 1. Values in a range close to 1 indicate the vortex core border and as mentioned above also the vortex axis. Hence, these values are used to mark the grid cell edges for refinement. For further simplification of the edge-cell marking we set all values larger than 0.98–1. Then we are able to apply our grid adaptation.

From the practical point of view, it is clear that the vortex core border cannot be well resolved in a single grid refinement step. Thus, in the first adaptation step the threshold for the sensor has to be explicitly lower than 1, but it can be gradually increased step by step.

5. APPLICATION AND RESULTS

In this section, we present the results of our grid adaptation algorithm by applying it to several examples.

In the first application, we consider an analytical flow in a cylinder to explain the main aspects of our approach. We start with a very coarse unstructured tetrahedral grid, on which we define the analytical vortex. The cylinder has a diameter of 16 m and a height of 2 m. The initial grid has an average grid cell edge of 0.05 m.

An impression of the spatial grid resolution can be gained from Figure 6, where the cell distribution is illustrated for one slice and the outer boundary of the cylinder.

As the basic flow field we use two simple Burgers vortices

$$\begin{aligned} v_r &= -ar \\ v_\varphi &= \frac{\kappa}{r}\left[1-\exp\left(-\frac{ar^2}{2\nu}\right)\right] \\ v_z &= az \end{aligned} \tag{12}$$

with the kinematic viscosity $\nu = 1.5\times 10^{-5}\,\mathrm{m}^2\,\mathrm{s}^{-1}$. The parameter a is set to $a = 0.2\,\mathrm{s}^{-1}$. In our test cases the vortex strength κ is varied to simulate co-rotating and counter-rotating vortices with different strengths and hence different interactions. We choose a distance of 10 m between the analytical middle points of the vortices.

DOI: 10.1002/fld

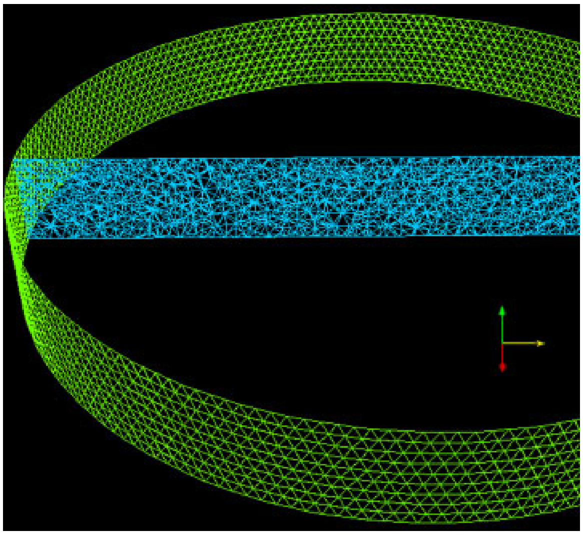

Figure 6. Slices through the unstructured tetrahedral grid.

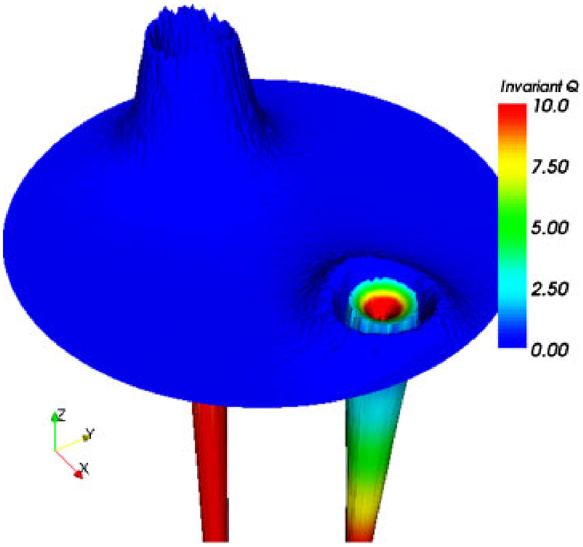

Figure 7. Velocity magnitude on a slice, grid structure elevated by the calculated vortex core adaptation sensor.

For the first vortex, the vortex strength is given by $\kappa = 6\,\mathrm{m}^2\,\mathrm{s}^{-1}$, while it is $\kappa = 4\,\mathrm{m}^2\mathrm{s}^{-1}$ for the second vortex. These values are indicating two co-rotating vortices with comparable strength. The numerically calculated vortex axes have hardly shifted in comparison with the predefined centers of the analytical vortices. Figure 7 shows one slice planar through the grid, color coded by the

DOI: 10.1002/fld

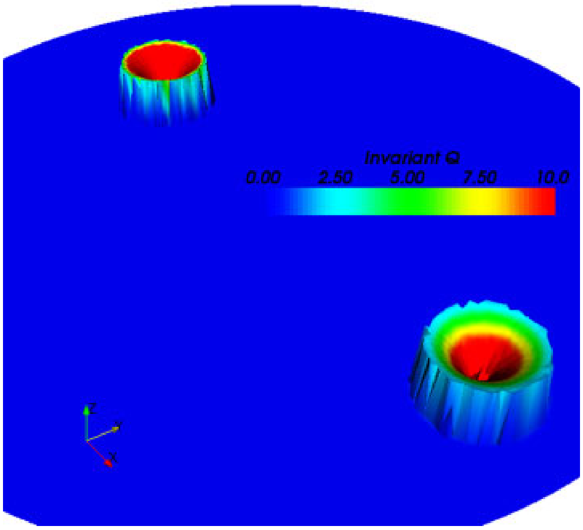

Figure 8. Co-rotating vortices, cutting plane elevated by vortex core adaptation sensor values, ridge line points have values of 1, additional color coding by the invariant Q of the velocity tensor.

invariant Q of the velocity gradient tensor and elevated by the values of our vortex core adaptation sensor. Further details are shown in Figure 8.

The given ring-like structure of both vortices is well represented and looks like the structure of a volcano crater. Moreover, as pointed out above, the Q values are significantly different from zero at the ridge line of the crater. However, the spatial resolution is obviously low which is revealed by the spiky contour of the raised cells.

Nonetheless, the sensor delivers the same strong signal for both vortices for cell refinement by values close to 1. The result of the following cell adaptation step is depicted in Figure 9. The locally refined zone is clearly visible as a ring-shaped crater lake. In this figure, to give a better impression of the vortex core, those cells are excluded, which are associated with the vortex axes.

In the second case, we consider two counter-rotating vortices. Thus, we set the vortex strength to $\kappa = 40\,\mathrm{m}^2\mathrm{s}^{-1}$ for the first vortex, and to $\kappa = -8\,\mathrm{m}^2\mathrm{s}^{-1}$ for the second vortex. The vortex strengths have a significantly different magnitude which leads to a strong interaction of these vortices. As a result, the stronger vortex shifts the vortex axis of the other vortex by 0.55 m outwards while its axis moves only by 3 cm.

Figure 10 gives an impression of the local flow structure. Here we choose the vortex definition based on the invariant Q again to highlight the asymmetry of the secondary vortex.

Figure 11 shows the results analogous to Figure 8; here, the calculated vortex core adaptation sensor is used to elevate the vortex core. In contrast to the definition by the invariant Q, our algorithm finds the core border very precisely even in this highly asymmetric case. However, in our algorithm it is important to take the shifts of the vortex axes also into account. Using the original axes that are employed to initiate the velocity field leads naturally to wrong results. This can be explained by analyzing the vorticity flux rate in the vortical regions.

DOI: 10.1002/fld

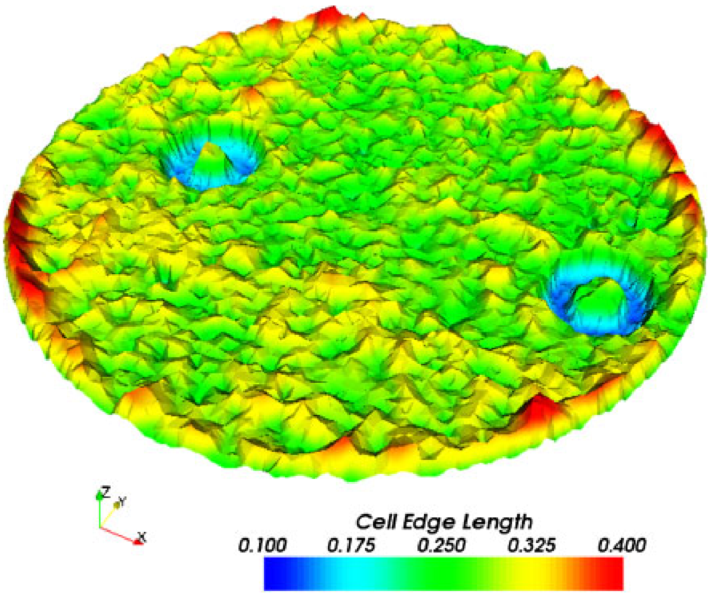

Figure 9. Co-rotating vortices, cell edge lengths after local grid refinement in cut plane, elevation scaled by 5 times the edge lengths.

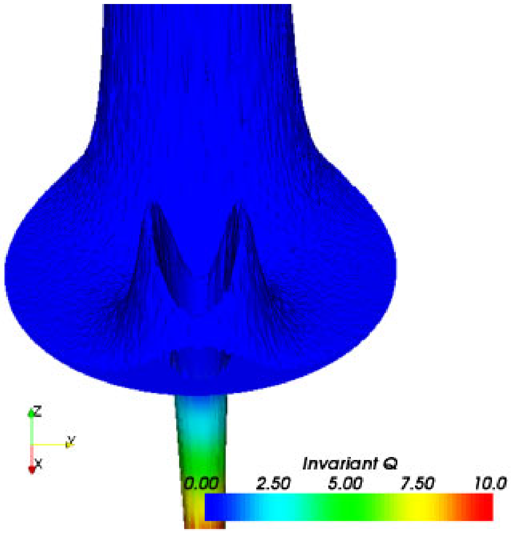

Figure 10. Vortical structures of two counter-rotating vortices with strongly different vortex strengths.

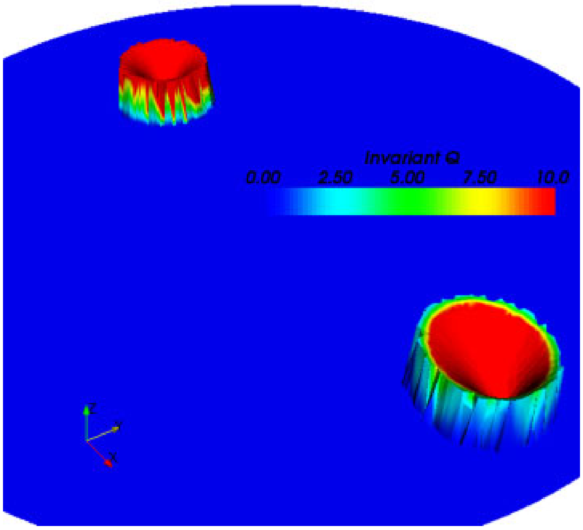

Figure 11. Counter rotating vortices, cutting plane elevated by vortex core adaptation sensor values, ridge line points have values of 1, additional color coding by the invariant Q of the velocity tensor.

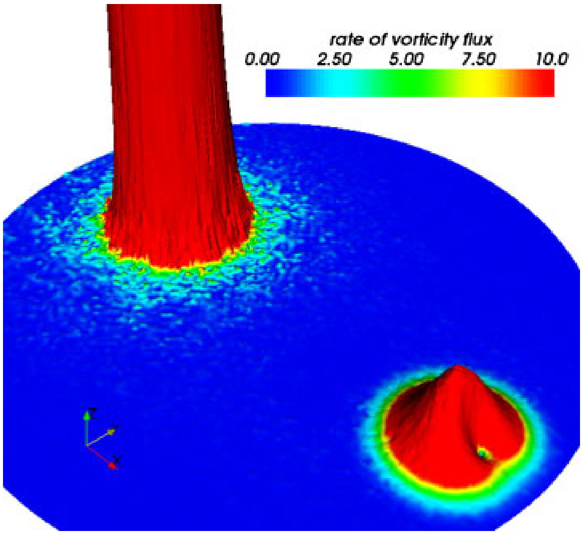

Figure 12. Elevation of the vorticity flux rate on a cutting plane.

Figure 12 shows the effect of the vortex interaction on the vorticity flux rate. For the weaker vortex the asymmetric location of the vortex axis correlates with the spot-like decline of the vorticity flux rate. Despite this asymmetry, we receive a similar result as in Figure 9 when we apply our adaptation module, see Figure 13.

Again, we used the cell edge lengths to visualize the refined vortex core border grid region.

DOI: 10.1002/fld

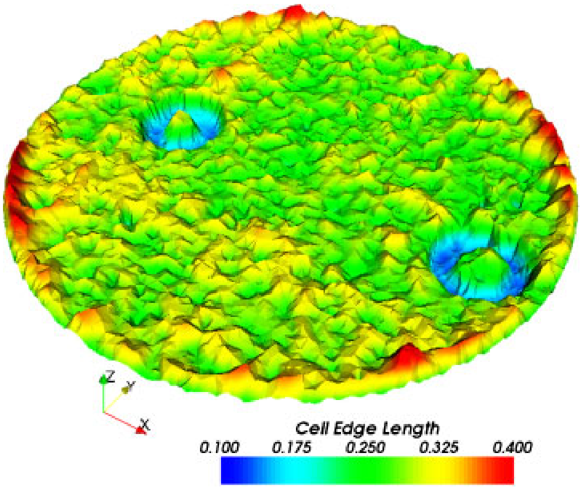

Figure 13. Counter-rotating vortices, cell edge lengths after local grid refinement in cut plane, elevation scaled by 5 times the edge lengths.

Next, we want to concentrate on the more realistic case of a pitching delta wing, which has been numerically computed. The flat, 65° swept, generic prismatic delta wing has a chord length of 1 m and a height of 5 mm. The delta wing has sharp edges with an edge angle of 15°. On account of this, the leading edge separation of the flow is exactly predefined and a wavy vortex is suppressed. We performed an *unsteady Reynolds-averaged Navier–Stokes* simulation on a Linux cluster using the DLR finite volume solver TAU for compressible flow [32], which has a second-order spatial and temporal discretization. The computations were performed using a highly pre-adapted [1], hybrid grid, which consists of 3 million points and 11 million cells, 8.8 million tetrahedra and 3 million prisms in 24 layers. The on-flow Mach number was 0.2 and the Reynolds number was 1.2 millions. For turbulence modeling, the Spalart–Almaras turbulence model with the Edwards modification was used, which shows adequate results. Time accurate simulations were performed with a time step of 0.0001 s. The angle of attack was varied from 16 to 24°. The simulation has been sped up by flux splitting to perform inner sub-iterations. In addition, a 3-v multi-grid cycle for solving the linear systems has been performed to further accelerate the computations. As another goal of our investigation was to analyze the symmetric or asymmetric behavior of the vortical delta wing flow, the assumption of symmetry in a plane could not be used and therefore, the flow field of the full delta wing configuration had to be computed. The flow field of such a symmetric delta wing, which is typical of modern fighter aircrafts, consists of large-scale vortical structures on its lee side. Typically, there are two primary vortices and smaller secondary, tertiary, and other subtype vortices, which are strongly interacting. Normally, the primary vortices are dominating the overall flow structure yielding the main part of the additional lift apart from the potential lift of the attached flow, see the sketch in Figure 14.

For the rest of this example, we will concentrate on an angle of attack of 24°. Although, at this angle of attack, the vortex breakdown phenomenon occurs, which results eventually in a vortex breakdown bubble, see Figure 15, we are particularly interested in resolving the stable flow field in front of the breakdown flow structures.

DOI: 10.1002/fld

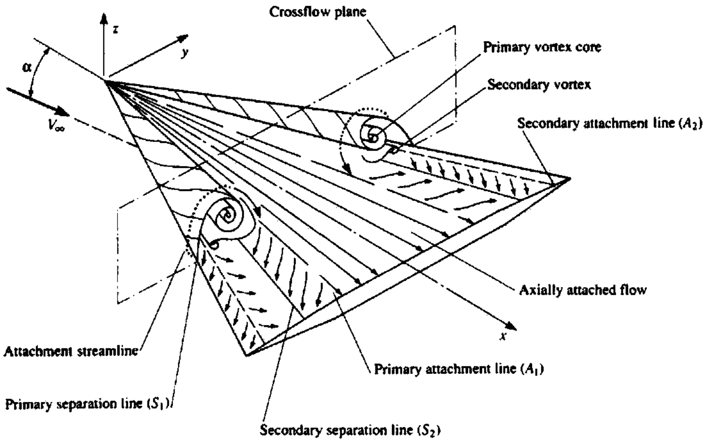
Crossflow plane
Primary vortex core
Secondary vortex
Secondary attachment line (A_2)
z
y
α
V_∞
Axially attached flow
x
Attachment streamline
Primary attachment line (A_1)
Primary separation line (S_1)
Secondary separation line (S_2)

Figure 14. Sketch of the vortical flow field above a pitching delta wing.

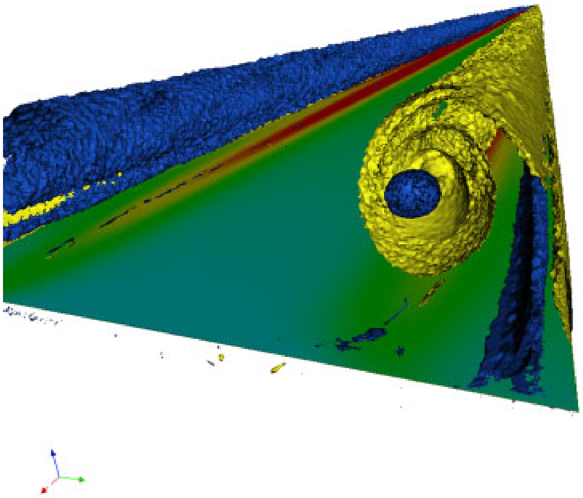

Figure 15. Vortex breakdown bubble visualized by iso-surfaces of the normalized helicity density criterion.

Therefore, we decided to reduce the grid region, where we want to apply our algorithm. Thus, we choose a representative region, where the dominant primary and secondary vortices are stable and no vortex breakdown occurs. The cutout-box is depicted in Figure 16.

For this region, we have generated a tetrahedral mesh with nearly isotropic grid cell sizes, on which we interpolated the flow data and to which we applied our adaptation algorithm. The principle flow structure within this box is illustrated in Figure 17, which shows the color-coded normalized helicity distribution on a slice highlighted by the vortex Q criterion. Additionally, the vortex axis for the primary and secondary vortices are depicted.

DOI: 10.1002/fld

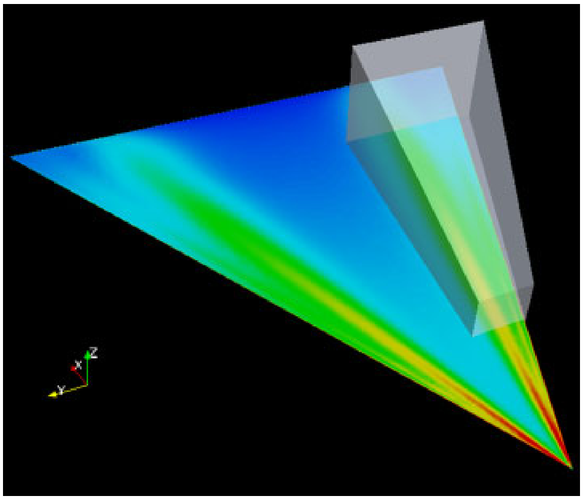

Figure 16. Cutout-box of the grid adaptation region.

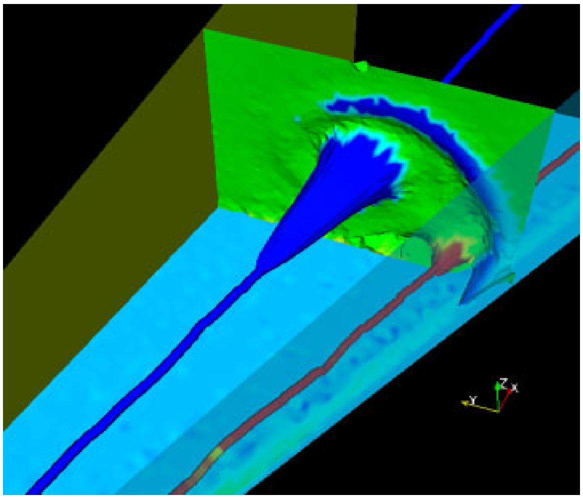

Figure 17. Principal flow structure in the cutout subregion, normalized helicity distribution on a slice highlighted by the vortex Q criterion, primary (blue) and secondary (red) vortex axis.

To compare our proposed algorithm to a vortex-feature algorithm based on scalar values, we first adapt the grid using the vortex Q criterion as the adaptation indicator. Figure 18 shows the results for this case. The normalized helicity density distribution on a slice has been color coded and amplified by the average cell edge length. The grid adaptation leads to a region with smaller cell sizes, which are less amplified than unrefined cells.

Besides the expected finer resolution of the primary and the secondary vortices, using the vortex Q indicator leads also to the additional refinement of the vortex shedding flow structure. In contrast

DOI: 10.1002/fld

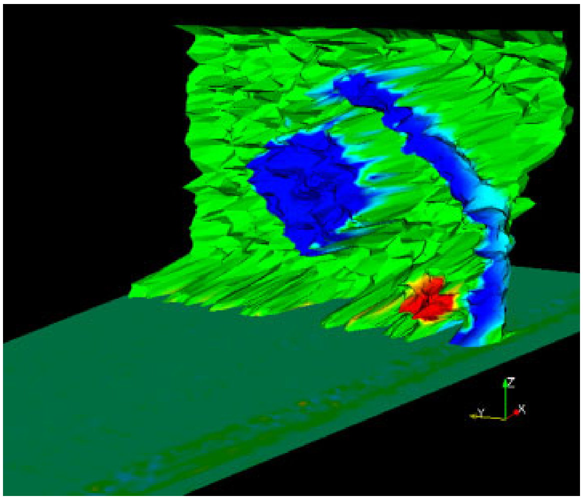

Figure 18. Cut through the adapted grid, vortex Q used as adaptation indicator, color-coded normalized helicity density distribution warped by the middle cell edge length.

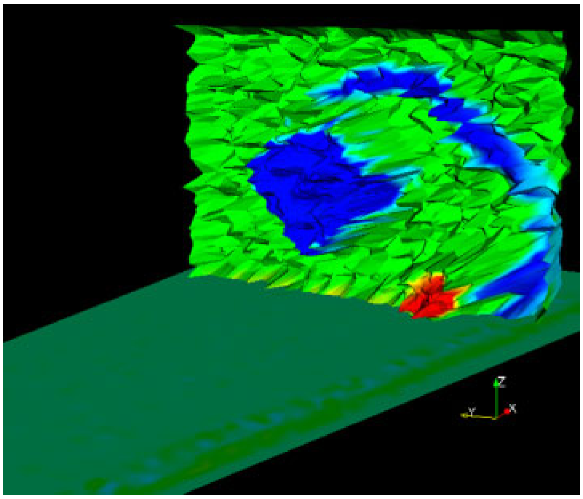

Figure 19. Cut through the adapted grid, vortex feature-based adaptation, color-coded normalized helicity density distribution warped by the middle cell edge length.

to the vortex Q indicator-based adaptation our proposed algorithm is able to distinguish between the shedding flow structure and the vortices and only refines the vortical region. Figure 19 contains the results of our algorithm, using the same visualization techniques as in Figure 18.

Obviously, our proposed algorithm can be used to refine vortex-related grid regions selectively, showing one remarkable advantage of our method. Moreover, the refined region is slightly smaller

DOI: 10.1002/fld

than the refined region in the first case. Finally, note that cells crossed by the vortex axis are also refined, which leads to a nearly identical grid structure in the centers of the vortices.

6. CONCLUSION

In this paper, we have proposed a feature-based vortex core border grid adaptation technique, which is based on the idea of finding the maximum of the circumferential velocity. After introducing well-known vortex definitions and criteria, we have outlined our algorithm in detail focusing on those vortex features necessary for our adaptation indicator. In particular, our vortex axis detection and our vortex core border calculation are crucial for our adaptation approach.

Another important point concerned the principles of our cell handling, in particular the edge refinement algorithm and the derivation of new cells. To this end, we described the cell development step by step, which allows the reader to understand our usage of the adaptation indicator. Since we think that also a de-refinement should be part of an adaptation strategy to allow a development of the flow solution and to avoid a freezing of an underdeveloped solution, we sketched our implementation of cell de-refinement.

After that, we concentrated on simple test cases of co-rotating and counter-rotating vortices. They demonstrated that our algorithm is able to detect and refine the grid region of the vortex core border even when the interaction of vortices leads to distorted vortical structures. Furthermore, we have shown that our algorithm is able to distinguish between real vortices and vortex sheets. Finally, we have shown that our algorithm does not only work for simple test cases but also for realistic problems.

However, so far we purely concentrated on the geometric topology of the grid and used a given flow solution to calculate our adaptation indicator. Thus, we have to remark that this is only one side of the problem of resolving flow field features. The other side is the flow solution itself. In particular, the roughness of the solution has to be considered in future work on grid adaptation. We intend to develop algorithms for identification and reconstruction of vortical flow features under the circumstances of a bad spatial resolution.

REFERENCES

1. Alrutz T, Rütten M. Investigation of vortex breakdown over a pitching delta wing applying the DLR TAU-Code with full, automatic grid adaptation. *Thirty-fifth AIAA Fluid Dynamics Conference*, Toronto, 6–9 June 2005; Paper 5162.
2. Lugt HJ. The dilemma of defining a vortex. In *Theoretical and Experimental Fluid Mechanics*, Müller U, Roesner KG, Schmidt B (eds). Springer: New York, 1979; 309–321.
3. Robinson SK. Coherent motions in the turbulent boundary layer. *Annual Review of Fluid Mechanics* 1991; **23**:601–639.
4. Cucitore R, Quadrio M, Baron A. On the effectiveness and limitations of local criteria for the identification of a vortex. *European Journal of Mechanics B/Fluids* 1999; **18**:261–282.
5. Saffman PG. *Vortex Dynamics*. Cambridge University Press: Cambridge, 1992.
6. Levy Y, Degani D, Seginer A. Graphical visualization of vortical flows by means of helicity. *AIAA Journal* 1990; **28**(8):1347–1352.
7. Dallmann U. Three-dimensional vortex structures and vorticity topology. *Proceedings of the IUTAM Symposium on Fundamental Aspects of Vortex Motion*, Tokyo, Japan, 1987.
8. Chong MS, Perry AE, Cantwell BJ. A general classification of three-dimensional flow fields. *Physics of Fluids A* 1990; **2**(5):261–265.

DOI: 10.1002/fld

9. Dallmann U. Analysis of simulations of topologically changing three-dimensional separated flows. *IUTAM Symposium on 'Separated Flows and Jets'*, Novosibirsk, 1990.
10. Hunt JRC, Wray AA, Moin P. Eddies, streams and convergence zones in turbulent flows. *Center of Turbulence Research Report CTR-S88*, 1988; 193–208.
11. Truesdell C. *The Kinematics of Vorticity*. Indiana University: Bloomington, IN, U.S.A., 1954.
12. Jeong J, Hussain F. On the identification of a vortex. *Journal of Fluid Mechanics* 1995; **285**:69–94.
13. Wu JZ, Xiong AK, Yang YT. Axial stretching and vortex definition. *Physics of Fluids* 2005; **17**:038108.
14. Chakraborty P, Balachandar S, Adrian RJ. On the relationships between local vortex identification schemes. *Journal of Fluid Mechanics* 2005; **535**:189–214.
15. Haller G. On objective definition of a vortex. *Journal of Fluid Mechanics* 2005; **525**:1–26.
16. Banks D, Singer B. Vortex tubes in turbulent flows: identification, representation, reconstruction. *Proceedings of IEEE Visualization '94*, Washington, DC, U.S.A., 1994; 132–139.
17. Miura H, Kida S. Identification of central lines of swirling motion in turbulence. *Proceedings of International Conference on Plasma Physics*, Nagoya, Japan, 1996; 866–896.
18. Kenwright DN, Haimes R. Vortex identification—applications in aerodynamics: a case study. *Proceedings of IEEE Visualization '97*, Phoenix, AZ, U.S.A., 1997; 413–416.
19. Sujudi D, Haimes R. Identification of swirling flow in 3D vector fields. *Twelfth AIAA CFD Conference*, San Diego, CA, 1995; AIAA Paper 95-1715.
20. Roth M. Automatic extraction of vortex core lines and other line-type features for scientific visualization. *Ph.D. Thesis*, Swiss Federal Institute of Technology Zürich, 2000.
21. Peikert R, Roth M. The 'parallel vectors' operator—a vector field primitive. *Proceedings of IEEE Visualization '99*, San Francisco, CA, 1999.
22. Sahner J, Weinkauf T, Hege H-C. Galilean invariant extraction an iconic representation of vortex core lines. *Eurographics—IEEE VGTC Symposium on Visualization*, Leeds, U.K., 2005.
23. Jiang M, Machiraju R, Thompson D. Detection and visualization of vortices. *Proceedings of IEEE Visualization 2002*, Massachusetts, U.S.A., 2002; 413–416.
24. Rütten M. Vortex axis calculation by using vortex features. *Thirty-fourth AIAA Fluid Dynamics Conference and Exhibit*, Portland, OR, 2004; AIAA Paper 2004-2353.
25. Lugt HJ. *Introduction to Vortex Theory*. Vortex Flow Press, Inc.: Maryland, 1996.
26. Burgers JM. A mathematical model illustrating the theory of turbulence. *Advances in Applied Mechanics* 1948; **1**:171–199.
27. Alrutz T, Matthias O. Parallel dynamic grid refinement for industrial applications. In *Proceedings ECCOMAS 2006*, 5–8 September 2006, Wesseling P, Onate E, Periaux J (eds). Egmond aan Zee: The Netherlands, 2006.
28. Alrutz T. *MEGAFLOW—Numerical Flow Simulation for Aircraft Design Results of the Second Phase of the German CFD Initiative MEGAFLOW Presented During its Closing Symposium* DLR, Braunschweig, Germany, 10–11 December 2002, *Hybrid Grid Adaptation in TAU*, Chapter 7, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, vol. 89. Springer: Berlin, 2005; 109–116.
29. Alrutz T. Erzeugung von unstrukturierten Netzen und deren Verfeinerung anhand des Adaptationsmoduls des DLR-TAU-Codes. Diplomarbeit, Universität Göttingen, 2002.
30. Habashi WG, Dompierre J, Bourgault Y, Fortin M, Vallet M-G. Certifiable computational fluid dynamics through mesh optimization. *AIAA Journal* 1998; **36**(5):703–711.
31. Arya S, Mount DM. Approximate nearest neighbor searching. *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM Press: New York, 1993; 271–280.
32. Schwamborn D, Gerhold T, Kessler R. The DLR-TAU Code—an overview. *Proceedings of ODAS 99, ONERA-DLR Aerospace Symposium* 21–24 June 1999, Paris, France, 1999.

DOI: 10.1002/fld