STEVE – Space-Time-Enclosing Volume Extraction

B.R. Schlei*
GSI Helmholtz Centre for Heavy Ion Research GmbH, Planckstraße 1, 64291 Darmstadt, Germany

In this paper, we describe a novel algorithm for the extraction of iso-valued hypersurfaces that may be implicitly contained in four-dimensional (4D) data sets, e.g., like time-varying computed tomographic three-dimensional images. This so-called STEVE (Space-Time-Enclosing Volume Extraction) algorithm is based on its three main ingredients: (i) an indexing scheme for 4D points specifically located within the tesserae-like $2 \times 2 \times 2 \times 2$-neighborhood of toxels (i.e., – without loss of generality – time-varying voxels), (ii) a vector path table that provides proper links for contributing volume segments, and (iii) a connectivity diagram that helps to resolve topological ambiguities. Any final hypersurface that will be generated by STEVE is guaranteed to be free from accidental rifts, i.e., it always fully encloses a region in the 4D space under consideration. Furthermore, the information of the interior/exterior of the enclosed regions is propagated to each of the tetrahedrons, which are embedded into 4D and which in their union represent the final, iso-valued hypersurface(s).

Keywords: surfaces and interfaces, computed tomography, computational techniques; simulations
PACS: 68.00.00, 87.57.Q- , 02.70.-c

I. INTRODUCTION

The following presentation is subject to a pending patent application (cf., Ref. [1]).
Recent advances in the field of computed tomography (cf., e.g., Refs. [2, 3]) have made available high-quality 4D (i.e., three spatial dimensions (3D) and one temporal dimension (+1D)) reconstructed sets of measured time-varying voxel data. Similarly structured data may be generated in other scientific fields also, e.g., in the field of theoretical heavy-ion physics. There, so-called fireballs (i.e., extremely hot zones of strongly compressed nuclear matter that are formed by nuclear collisions) expand and cool down while emitting subatomic particles [4]. For the proper fireball expansion modeling, relativistic fluid simulation codes are employed (cf., e.g., Refs. [4, 5]), which generate various – real-valued – field quantities (e.g., temperature, density, energy density, etc.) on a cartesian spatial (3D) grid for the discretized (+1D) time, i.e., at fixed time steps.

The 4D history data of the fireball simulation can be used by a theoretician to calculate various observables (e.g., subatomic particle production rates). In doing so, it is very often necessary to determine an isotherme (cf., e.g., Refs. [6, 7]), i.e., a manifold of codimension 1 – or iso-hypersurface – at fixed temperature, which is implicitly contained in the discretized relativistic fluid history. If only so-called central heavy-ion collisions (i.e., the impact parameter of the collision equals to zero) are simulated, it is sufficient to model $2 + 1$D (i.e., 2D spatial cartesian coordinates: radius, $r$, and the beam axis position, $z$, of the incident nuclei; plus 1D time) relativistic fluids, because of the rotational symmetry of the system under consideration (cf., e.g., Refs. [8, 9]). Then the task of iso-hypersurface extraction reduces to the task of iso-surface extraction in 3D for voxelized data [12].

Direct 3D methods (i.e., those which do not extrude the data to higher dimensions for the purpose of iso-surface construction) can be subdivided basically into three classes (cf., Ref. [12] and Refs therein for more detail): those algorithms, which (i) use templates (cf., e.g., the Marching Cubes algorithm [13, 14]), (ii) introduce polarities at the voxel sites, and which implicitly solve spatial ambiguities (cf., e.g., the Marching Lines algorithm [13, 14]), and (iii) are protomesh-based, and which solve spatial ambiguities explicitly (cf., e.g., VESTA [12]). However, the majority of real heavy-ion collisions has an impact parameter that usually deviates strongly from zero. Hence, the rotational symmetry is absent (cf., e.g., Refs. [8, 17]), which may force a theoretician to perform a full-fledged 4D simulation of a heavy-ion collision with – perhaps – a subsequent numerical iso-hypersurface construction.

Similarly, one cannot always assume that 4D computed tomography data have internal symmetrical features (except, e.g., for the trivial case of temporally static data). The ability to extract iso-hypersurfaces from such (discretized) data may allow for an explicit, continuous 4D shape-representation (e.g., a continuous chronological evolution of 3D shapes). This subject of numerical iso-hypersurface extraction in 4D is not new (cf., below). As in the 3D case, in 4D there also seem to be three classes of direct methods to construct iso-hypersurfaces from voxel (i.e., time-varying voxel) data sets. For a template-based 4D algorithm, cf., e.g., Ref. [18]. An algorithm that generalizes the ideas of the 3D Marching Lines algorithm into 4D is provided by the much earlier work of Fidrich [19]. Here, however, we shall present in great detail – for the very first time – the protomesh-based “Space-Time-Enclosing Volume Extraction” algorithm (in the following shortly referred to as STEVE; cf., Ref. [20] for its first announcement).

*Email address: schlei@me.com
This paper is organized as follows. First, we shall discuss the initial data mesh features. In particular, we shall compare proper 2D, 3D, and 4D meshes and stress certain analogies, because we would like to compare a variety of features of STEVE eventually with its 2D and 3D counterparts, DICONEX [21] and VESTA [12], respectively. Next, the volume extraction framework will be presented. In doing so, we are going to introduce for each 4D $2 \times 2 \times 2 \times 2$-neighborhood of toxels a proper indexing scheme and a complete, corresponding vector path table (cf., Table I, in the Appendix) that will provide all possible links for contributing volume segments. Particular emphasis will be put onto the treatment of topological ambiguities. The latter will make clear that more than one solutions are generally possible if one is faced with the task of iso-hypersurface construction in discretized 4D spaces. In an application related section, we shall explore projections of constructed hypersurfaces. Finally, this paper will conclude with a short summary.

II. INITIAL MESH FEATURES

In the following, we shall take a closer look at the geometrical structure of the underlying meshes (or grids) in 2D, 3D, and 4D, which support the data that should be processed. In particular, we shall consider here only homogeneous, $N$-cubical (cartesian) grids, with $N$ referring to the integral dimension of the space under consideration. In doing so, we shall use two different approaches. Either, we shall look at the data globally, or we shall look at a $2(x)2^{N-1}$-neighborhood of the data. In the following, we shall use in the first case the term “global view” (GV), and in the latter case we shall use the term “neighborhood view” (NV), respectively.

A. Pixels, Voxels, and Toxels

In Fig. 1, the various coordinate systems that we are going to use ($x|y$ in 2D, $x|y|z$ in 3D, and $x|y|z|t$ in 4D, respectively) are shown together with samples of single, corresponding picture elements (in GV). The centers of the elements are each marked with a sphere. In Fig. 1.a, we show a pixel (i.e., picture element) that is represented by a square, which is surrounded by its four edges. In Fig. 1.b, we show a voxel (i.e., volume pixel) that is represented by (the projection of) a 3D cube, which is surrounded in 3D by its six squares. In Fig. 1.c, we show a toxel (i.e., – without loss of generality – time-varying voxel) that is represented by (the projection of) a 4D cube, and is surrounded in 4D by its eight cubes. Note, that each of these picture elements are already perfectly enclosed by their surrounding shape elements. However, for an iso-valued contour, surface, or hypersurface, respectively, a different supporting point set is generally favored, than just the end points of the lines and/or the corner points of the surrounding shape elements themselves.

B. Protomesh Building Blocks

In the following, we shall prepare ourselves for the construction of iso-valued hypersurfaces while using a protomesh-based technique that is derived from the initially identified surrounding boundary structures. For the anticipated type of construction of contours in 2D and surfaces in 3D, we would like to refer the reader to the detailed descriptions of DICONEX [21] and VESTA [12], respectively. However, here we shall shortly review the basics that have led to both, DICONEX and VESTA, because we shall proceed analogously in 4D.

In Fig. 2, we show in GV pairs of picture elements in 2D, 3D and 4D, respectively, which are in direct contact. In each dimension, one element is active, i.e., it is marked for enclosure (indicated by the spheres), and the other element is considered inactive, i.e., it should
FIG. 3: Initial building blocks for contour, surface and hypersurface construction in (a) for 2D, in (b) & (c) for 3D, and in (d) – (f) for 4D, respectively (see text).

not be enclosed (no spheres are placed in their centers). Each center of an active element is the origin of a range vector, which ends in the center of the inactive neighboring picture element. The range vectors (dark gray) define the bounds within which a support point (black dots) of the final extracted, iso-valued manifolds may be positioned. The element pairs are separated in 2D (cf., Fig 2.a) by a single vector (black), which has the active pixel at its left, and in 3D (cf., Fig 2.b) by a single square with four dashed lines (light gray) connecting the center of the square (black dot) with the middle points (light gray dots) of its four edges. In 4D (cf., Fig 2.c) the element pairs are separated by a single cube with six dashed lines (light gray) connecting the center of the cube (black dot) with its face centers (light gray dots), i.e., its six surrounding squares.

These transitions from an active to an inactive picture element are of particular importance, because the final manifolds of codimension 1 should be located in their vicinity. The transitions are represented by the geometrical structures that are described above (cf., Fig 2). These structures form the initial building blocks (cf., Fig 3) for the algorithms under consideration. Regardless of the dimension of the space, the centers (black dots) of the building blocks are going to be the support points of the final shape-enclosing manifolds. The light gray dots indicate the points of contact, where a building block will connect to another one when the manifolds are constructed. The range vectors (dark gray) define the bounds for the final support points (black dots), and they also indicate the interior of the shapes, which ought to be enclosed, through their orientation.

The building blocks are shown (each in GV) in Fig 3.a, for 2D as implemented in the DICONEX algorithm; in Fig 3.b, for 3D as implemented in VESTA (without internal vector paths); in Fig 3.c, for 3D as implemented in VESTA with vector paths; in Fig 3.d, for 4D as implemented in the STEVE algorithm (without internal vector paths); in Fig 3.e, for 4D as exploded view for positive orientation (⊕, i.e., the range vector points into positive x-, y-, z-, or t-direction, respectively) with the vector paths as implemented in Table I (cf., the Appendix); in Fig 3.f, for 4D as exploded view for negative orientation (⊖, i.e., the range vector points into negative x-, y-, z-, or t-direction, respectively) with the vector paths as implemented in Table I (cf., the Appendix). Note that in Fig.s 3.e and 3.f, the tiny black arrows mark vector pairs, which belong together.

III. THE VOLUME EXTRACTION FRAMEWORK

Eventually, the properly connected sets of building blocks, which have been introduced in the previous subsection, form the protomeshes that will be processed further. For the 2D and 3D processing, we would like to refer the reader to Ref.s [21] and [12], (i.e., the detailed descriptions of DICONEX and VESTA, respectively). In this section, however, we shall describe – among many other things – how STEVE processes the identified initial boundary volumes that are embedded in 4D spaces.

A. Indexing Scheme and Vector Paths

It is sufficient (particularly here, in 4D) to discuss all transitions from active to inactive toxels within a $2 \times 2 \times 2 \times 2$-neighborhood (or 4D-cell), because here we deal only with homogeneous, tesseract-like 4D grids. Such an approach has the clear advantage that we can label each relevant 4D point of the 4D-cell with an unique index.

In Fig. 4, we show the notations, i.e., identities (IDs), which we have chosen for a given 4D-cell, in NV. In Fig. 4.a, all sixteen toxel site IDs are shown. No.s 0 through 7 represent the “past,” whereas no.s 8 through 15 represent the “future,” while using – without loss of generality – time as the fourth dimension. In Fig. 4.b, all thirty-two potential boundary cube centers (i.e., final manifold support points for a single 4D-cell) are indicated. And finally, in Fig. 4.c, the all twenty-four connectivity points (i.e., potential points of ambiguity) for neighboring boundary cubes are shown. As an example, Fig. 4.d, shows for toxel no. 0 four range vectors (with positive (i.e., ⊕) orientations each in the x-, y-, z-,
FIG. 4: Indexing scheme for a $2 \times 2 \times 2 \times 2$-neighborhood of toxels; (a) toxel site IDs; (b) boundary cube centers; (c) connectivity points; (d) range vectors for toxel no. 0, superimposed with corresponding boundary cube centers.

and $t$-directions), superimposed with their corresponding boundary cube centers. Note that the central cubes in Fig.s 4.b–d represent the “present.”

As Fig. 4.d already suggests, each of the sixteen toxels may contribute within the 4-cell with exactly four boundary volume octants (i.e., an eights of a full boundary volume). Furthermore, each boundary volume octant is represented within STEVE by a triplet of vector pairs, which each connects a connectivity point through the corresponding boundary cube center to another connectivity point (cf., Fig.s 3.e and 3.f). In total, one has $16 \times 4 \times 3 = 192$ different possible vector paths within a given toxel neighborhood. The indexing scheme as shown in Fig. 4 and the table of the 192 vector paths (cf., Table I, in the Appendix) form two of the three main ingredients that STEVE uses for the purpose of iso-hypersurface construction.

The third main ingredient for STEVE is a connectivity diagram (cf., Fig. 13.c) that helps to resolve topological ambiguities, when connectivity points will become points of ambiguity. The latter will be discussed in subsection III.G.

B. Generation of a single Tetrahedron

In this subsection, we shall demonstrate the extraction of an iso-hypersurface for a single toxel (cf., Fig. 1.c).

In 2D, a single pixel is represented within a $2 \times 2$-neighborhood by a quarter of its (quadratic) area. In 3D, a single voxel is represented within a $2 \times 2 \times 2$-neighborhood by an eights of its (cubic) volume. Analogously in 4D, a single toxel is represented within a $2 \times 2 \times 2 \times 2$-neighborhood by a sixteenth of its (4-cubic) space-time (if we choose three spatial and one temporal dimensions). Hence, we need sixteen different 4-cells for the proper construction of the complete hypersurface of a single toxel. In order to demonstrate how STEVE uses the previously introduced indexing scheme (cf., Fig. 4) in combination with the vector path table (cf., Table I, in the Appendix), we shall process here a single sixteenth of a toxel.

As an example, we shall consider within a 4-cell the four boundary cubes for the single toxel at site ID no. 7 (cf., Fig. 5.a). These are the volumes no. 11 $\oplus$ (in the positive $x$-direction; cf., Fig. 5.b), no. 9 $\ominus$ (in the negative $y$-direction; cf., Fig. 5.c), no. 6 $\ominus$ (in the negative $z$-direction; cf., Fig. 5.d), and no. 18 $\oplus$ (in the positive $t$-direction; cf., Fig. 5.e). In total we obtain 24 initial vectors (cf., Fig. 6.a), which – in a first step – can be combined into the four initial cyclic vector paths: $37 \rightarrow 11 \rightarrow 36 \rightarrow 6 \rightarrow 34 \rightarrow 9 \rightarrow 37$, $36 \rightarrow 11 \rightarrow 49 \rightarrow 18 \rightarrow 44 \rightarrow 6 \rightarrow 36$, $47 \rightarrow 18 \rightarrow 49 \rightarrow 11 \rightarrow 37 \rightarrow 9 \rightarrow 47$, and $34 \rightarrow 6 \rightarrow 44 \rightarrow 18 \rightarrow 47 \rightarrow 9 \rightarrow 34$, respectively.

Note that at the connectivity points, vectors are only connected such that the end-point of a predecessor connects to the starting-point of a linked successor (which never must be anti-parallel to the preceeding vector). The vector connectivities at the boundary cube centers are predefined through the vector path table (cf., Table I, in the Appendix).

FIG. 5: (a) A sixteenth of a toxel within a 4-cell at site no. 7; detailed views of the corresponding boundary volumes (b) no. 11 $\oplus$; (c) no. 9 $\ominus$; (d) no. 6 $\ominus$; (e) no. 18 $\oplus$ (cf., Table I).
FIG. 6: Continuation of Fig. 5: (a) the four initial cyclic vector paths within a 4-cell. The final tetrahedron together with the final cyclic vector path (b) $11 \rightarrow 6 \rightarrow 9 \rightarrow 11$ (embedded into the 4D-cell); (c) $11 \rightarrow 18 \rightarrow 6 \rightarrow 11$; (d) $18 \rightarrow 11 \rightarrow 9 \rightarrow 18$; (e) $6 \rightarrow 18 \rightarrow 9 \rightarrow 6$. (f) The final tetrahedron, where each of its edges represents a pair of anti-parallel vectors.

In a second step, all connectivity points (i.e., those with point IDs above 31; cf., Fig. 4.c) will be discarded. As a result, one obtains a single tetrahedron that is represented by its four final, reduced cyclic vector paths (cf., Fig.s 6.b–f). Each of these cyclic vector paths represents a triangle, which is embedded into 4D. Note that the initial orientations of the boundary volumes have been passed on, such that a consistent evaluation of 4-normal vectors is possible (cf., e.g., Ref.s [22, 23]).

If we repeat the above processing for the remaining fifteen toxel sites within the 4-cell, we end up in total with sixteen oriented tetrahedrons that represent in their union the final STEVE-hypersurface for a single, isolated active toxel (cf., Fig. 7.c). Note that in 4D the orientations of the surface elements (i.e., triangles) of the tetrahedrons are not drawn, because each triangle is oriented both ways in the figure. For comparison we also show in Fig. 7.a the DICONEX-contour (consisting of four oriented edges, i.e., vectors) for a single, isolated pixel [21]. And in Fig. 7.b, we show the VESTA-surface (consisting of eight oriented triangles) for a single, isolated voxel [12]. Note that in the 3D case, each edge of the octahedron in the figure represents two anti-parallel vectors.

C. Isochronous Hypersurface Sections

While considering time as the fourth dimension, a so-called isochronous hypersurface section (cf., Ref. [17], for an application) can be generated, which also has a very simple geometry within a given 4-cell (i.e., similar to the generated single tetrahedrons in the previous subsection). If all eight toxel sites of the “past” are active, and simultaneously all eight toxel sites of the “future” are inactive (cf., Fig. 8), or vice versa (cf., Fig. 9), then one obtains as a result a cube-shaped element (as shown in the figures at the intermediate time, here, the “present”), which simply fills the whole 3D space at the fixed time, $t$.

FIG. 8: Generation of an isochronous hypersurface section within a 4-cell: (a) initial boundary cubes; (b) initial cyclic vector paths; (c) final cube-shaped hypersurface element, with one final cyclic (clockwise oriented) vector path indicating the orientation of the hypersurface section.
FIG. 9: As in Fig. 8, but with an inverted toxel site occupancy: (a) initial boundary cubes; (b) initial cyclic vector paths; (c) final cubic shaped hypersurface element, with one final cyclic (counterclockwise oriented) vector path indicating the orientation of the hypersurface section.

Note that in each case, the initial eight boundary cubes yield after consideration of both, the indexing scheme (cf., Fig. 4) and the vector path table (cf., Table I), six initial cyclic vector paths. The orientations of the final vector paths are inherited from the initial ones (i.e., either \( \oplus \) or \( \ominus \); cf., Figs. 3.e and 3.f, and Table I) after the removal of the connectivity points. In Figs. 8.b and 9.b, the tiny black arrows mark the initial vectors, which lead to the final vector 4-cycles, which are pronounced in Figs. 8.c and 9.c, respectively.

D. Subspaces and Bounding Shapes

In the previous two subsections, we have encountered two rather simple 3D shapes as resulting hypersurface sections, i.e., a tetrahedron and a cube, respectively. The 4-cells within which STEVE determines iso-hypersurface sections, each decompose into eight 3-cubes (or cubes), i.e., 3D subspaces. Both previously determined hypersurface sections, i.e., the tetrahedron and the cube, are bounded by either four triangles (i.e., 3-cycles; cf., Fig. 10.c) or six squares (i.e., 4-cycles; cf., Fig. 10.e), respectively, which themselves are embedded into the corresponding 3D subspaces.

In fact, all possible hypersurface sections that can be generated with the STEVE algorithm will be bounded by the \( N \)-cycles (\( N = 3, 4, 5, 6, 7, 8, 9, 12 \)), which VESTA [12] would create, if it were processing the active voxels within properly arranged 3-cells (i.e., \( 2 \times 2 \times 2 \)-neighborhoods of voxels). This observation agrees with the fact that the number of vector paths (i.e., 192) for the STEVE algorithm (cf., Table I, in the Appendix) equals to eight times of the number of vector paths (i.e., \( 8 \times 24 \)) for the Marching VESTA (cf., Ref. [12], Table 1).

In a lower-dimensional analogy, the DICONEX [21] algorithm determines all (properly oriented) line segments within the cube-bounding 2D subspaces, i.e., its six squares. In Fig. 10, the complete tiling sets of sections of the manifolds of codimension 1 for 2D and 3D (sub)spaces as determined by the DICONEX, VESTA, and STEVE algorithms are depicted. Note that multiple sections could be generated within in a given (sub)space (for more detail, cf., Ref. [12]).

E. Decomposition of Hypersurface Sections

Due to the various 3D bounding shapes as shown in Figs. 10.c–10.p, the hypersurface sections which STEVE computes could be very complex-shaped polyhedrons (that are embedded into 4D). For visualization purposes, or, e.g., for the purpose of 4-normal vector determination (cf., e.g., Refs. [22, 23]) it may be desirable to decompose the polyhedrons into a set of tetrahedrons, because these simplices are better suited for the considered tasks.

As an example, we show in Fig. 11, how to decompose a single cube into twenty-four tetrahedrons. In Fig. 11.a, a cube is shown, which consists of six 3D tilings (4-cycles) as shown in Fig. 10.e. In Fig. 11.b, for each 4-cycle, its center of mass point (face center) is determined. In Fig. 11.c, the surface cycles are decomposed into triangles, while connecting the face centers with the corre-
FIG. 11: Decomposition of a cube into twenty-four tetrahedrons (see text).

sponding 4-cycle support points. Note that triangles will not be decomposed any further. Furthermore, each newly drawn line actually represents a pair of anti-parallel vectors.

Next, in Fig. 11.d, all triangles, which enclose a particular single volume are collected into an object along with the information, to which \( N \)-cycles the triangles belong: for all face centers of each of the found objects, the absolute center of the enclosed volume is determined. In Fig. 11.e, lines are introduced that connect the face centers with the absolute volume center. Finally, in Fig. 11.f, the additional connections of all 4-cycle support points with the absolute volume center finally yields the twenty-four properly oriented tetrahedrons.

Within STEVE, this approach has been extended to the more or less complex shaped polyhedrons. Note that polyhedrons, which are embedded into 4D, could be decomposed while using fewer tetrahedrons also. E.g., one could decompose a cube into just five tetrahedrons. However, one should be aware, that such a choice may introduce directional ambiguities into the space-time considered here. For the remainder of this paper, we shall apply the variant as depicted in Fig. 11. As a result, we shall always obtain a directionally most unbiased decomposition. Note that single tetrahedrons will not be decomposed any further.

F. Generation of a Triangular Strut

In another example, we briefly demonstrate the generation of a hypersurface section that has the shape of a triangular strut. In Fig. 12, two toxels (in NV) at site IDs no. 7 and no. 15 are in contact through a single volume. This volume of contact is visible within the cube that represents here the “present” (cf., Fig. 12.a). Once again, the application of both, the indexing scheme (cf., Fig. 4) and the vector path table (cf., Table I), yields five initial cyclic vector paths (cf., Fig. 12.b). Fig. 12.c shows the final triangular strut-shaped hypersurface element, with two final cyclic vector paths indicating its orientation. Note that this hypersurface element is bounded by two 3-cycles, as shown in Fig. 10.c, and by three 4-cycles, as shown in Fig. 10.d, respectively. Finally, in Fig. 12.d, a decomposition of the hypersurface element into fourteen oriented tetrahedrons is shown as described in the previous subsection (cf., Fig. 11).

Until now, we did not encounter any particular topological ambiguities while constructing the heretofore discussed hypersurface sections. However, this is going to change in the next subsection.

G. Ambiguous Connectivity

The here considered discretized spaces could lead to topological ambiguities. In 2D this is the case, if two active pixels share only one common point (i.e., a vertex; cf., Ref. [21] for more detail). In 3D this is the case, if two active voxels share only one common edge (cf., Ref. [12] for more detail). Analogously in 4D this is the case, if two active toxels share only one common face (square).

In Fig. 13.a, we show a 4-cell with two active toxels at site IDs no. 4 and no. 15. The toxels are in contact through a single face. Within the cube that here represents the “present” this surface of contact is visible. In
Fig. 13: Encounter of a topological ambiguity within a 4-cell: (a) initial boundary cubes; (b) initial cyclic vector paths; (c) connectivity diagram.

Fig. 14: Continuation of Fig. 13: (a) two resulting tetrahedrons from the “disconnect” mode; (b) a final, more complex-shaped hypersurface element, resulting from the “connect” mode; (c) as in (d), but with a hypersurface section that is decomposed into sixteen tetrahedrons.

Fig. 15: Hypersurface section generation for two toxels, which are in contact through a single edge: (a) initial boundary cubes; (b) initial cyclic vector paths; (c) two resulting tetrahedrons.

H. Disjunct Hypersurface Sections

In 3D, VESTA never connects two voxels, which are in contact only through a single vertex (cf., Ref. [12] for
FIG. 16: Hypersurface section generation for two toxels, which are in contact through a single vertex: (a) initial boundary cubes; (b) initial cyclic vector paths; (c) two resulting tetrahedrons.

more detail). In 4D, we have a similar situation when considering toxels that are in direct contact.

In Fig.s 16 and 17, the process of hypersurface section generation is shown for two toxels that are in contact only through a single edge and through a single vertex, respectively. In both cases, we simply obtain two tetrahedrons as final hypersurface sections, since the initial cyclic vector paths form two disjunct sets with four cyclic paths each. Apparently, toxel pairs with such a weak connectivity will always result in two separate hypersurface segments. This concludes the technical section of this paper.

IV. CONTINUOUS EVOLUTION OF SURFACES

This section is more related to applications. Here, we shall discuss a few use cases for hypersurfaces (or hypersurface sections) that have been generated while applying STEVE to 4D data.

A. A single $2 \times 2 \times 2 \times 2$-Neighborhood

In a first example, we shall extract continuously evolving surfaces from a hypersurface segment, which has been generated with STEVE for a single 4-cell. In Fig. 17, we show a sequence of one and the same 4-cell (i.e., in NV) with six activated toxel sites each. In the “past” (left side) the four toxels with IDs no.s 4, 5, 6 and 7 are active, whereas in the “future” (right side) only the two toxels with IDs no. 12 and no. 14 are active (cf., Fig. 4.a). STEVE has been applied to this configuration in its global “disconnect” mode, in order to determine the corresponding hypersurface section (i.e., the networks of black lines in the figure). For each tesseract (except for the first and the last one), an additional intersecting cube is drawn for various fixed times $t$. In fact, the parameter, $t$, indicates the relative time (that ranges between the extremes of 0 and 1), where the hypersurface section has been intersected.

Fig. 17 shows a temporal evolution of slices (i.e., 3D surface segments) of the generated hypersurface segment, which has been properly decomposed into a set of connected tetrahedrons (cf., subsection III.E). Each tetrahedron – when intersected – could contribute to a time slice either with a single point, a single edge, a single triangle, a single quadrilateral, or the whole volume of the tetrahedron. Here it is demonstrated, how a single square may transform continuously and smoothly into two separate triangles. Hence, the depicted 4D hypersurface section

FIG. 17: Continuous transformation of a single quadrilateral into two separate triangles (see text).
FIG. 18: (Taken from Ref. [20].) Generation of chronologically evolving freezeout hypersurfaces: (a) & (h) selected grid points of the discretized 3D temperature fields at two subsequent time steps; (b) – (g) various time projections of the generated iso-hypersurface.

establishes a correspondence between the surfaces in the “past” and in the “future” (and between those anywhere in between).

B. Two consecutive 3D Data Sets

In this subsection, we provide lastly an example for freezeout hypersurface (FOHS) extraction from 3 + 1D hydrodynamic simulation data (cf., e.g., Ref. [9] and Ref.s therein). On a cartesian 3D grid, a relativistic fluid has been propagated numerically. The spheres in Fig.s 18.a and 18.h represent grid points above a certain threshold temperature at the two subsequent time steps (in GV) where the considered fireball decays into two pieces. In particular, STEVE has been used (in its global “disconnect” mode) to determine the (continuous) iso-thermal hypersurface that lies between the two shown 3D data sets, i.e., a FOHS (of fixed temperature).

In doing so, STEVE has made use of the range vectors (cf., e.g., Figs 2.c and 4.c) in order to determine a (with respect to linear interpolation) most correctly located (and thereby smoother) iso-hypersurface (which is not shown here). Note that STEVE provides interpolations for all other field values, which have been associated with each voxel, as well. For the purpose of FOHS, STEVE will not just be applied to two subsequent time steps as shown in the figure, but to the full set of 4D simulation data. The resulting total FOHS will allow for the further calculation of various observables [5–11].

In Fig.s 18.b – 18.g, various time projections of the chosen FOHS section are shown for visualization purposes. Note that (as it has been the case in the previous subsection) any finite number of chronologically evolving 3D surfaces can be extracted.

V. SUMMARY

In summary, the 4D protomesh-based iso-hypersurface construction algorithm STEVE has been described here for the very first time in great detail. Where 4D template-based approaches (cf., e.g., Ref. [18]) may require the storage of (due to possible topological ambiguities at least) \(2^{16} = 65,536\) volume tile templates, the STEVE algorithm basically requires the storage of 192 vector paths only.

We would like to stress that – similar to the cases of iso-contour and iso-surface extraction from 2D pixel and 3D voxel data sets, respectively (cf., DICONEX [21] and VESTA [12]), – more than one iso-hypersurface solutions are possible. This is simply a consequence of the discretization of the 4D spaces under consideration, where topological ambiguities will allow for two different global treatments (i.e., global “disconnect” or global “connect” connectivity modes), and/or an additional local treatment (i.e., local “mixed” connectivity mode). The proper treatment of possible ambiguities ensures that accidental rifts in the final hypersurfaces will be avoided by all means.

STEVE constructs iso-hypersurfaces from the grounds up. i.e., it continuously transforms the set of initially identified octants (i.e., eights of a cube) into the final set of tetrahedrons. This set represents the final hypersurface or a finite set of hypersurfaces. Note that the initially given information about the interior/exterior of the enclosed 4D regions will be propagated to the final result.

Finally, we would like to point out that this work represents – particularly because of the explicit definition of all 192 possible vector paths (cf., Table I, in the Appendix) – also a self-contained and complete instruction manual for iso-hypersurface extraction from homogeneous, tesseract-like structured 4D spaces, e.g., like time-varying computed tomographic 3D images.

VI. ACKNOWLEDGEMENT

Some initial work has been supported by the Department of Energy under contract W-7405-ENG-36. The particular funding has resulted in a first version of ANSI C-based software [24], where ambiguities are resolved globally only.

VII. APPENDIX

In 2D, a shape could be enclosed either with a clockwise or a counter-clockwise contour. The DICONEX [21] algorithm uses the latter, i.e., shapes will always be enclosed with counter-clockwise contours. Similarly, in 3D the normal vectors of a shape-enclosing surface could either point to the interior or to the exterior of the enclosed shape. VESTA [12] uses the latter, i.e., normal vectors of shape-enclosing surfaces always point to the shapes’ exterior. Using a right-hand rule, this has direct consequences for the orientations of the three vectors, which circumscribe triangles (i.e., the simplices that in their
For the orientations of the initial voxel face vectors (for a single rectangular surface tile; (b) in 4D, for a triangular strut-shaped hypersurface segment.

FIG. 19: Orientations of bounding elements: (a) in 3D, for a single rectangular surface tile; (b) in 4D, for a triangular strut-shaped hypersurface segment.

Analogously, one has in 4D also two choices for the orientation of the vector paths as shown in Table I. I.e., the first three path columns that here refer to $\ominus$, instead could have referred to $\oplus$; and the last three path columns that here refer to $\ominus$, instead could have referred to $\oplus$ (cf., Fig. 10.c). In Fig. 19, we visualize how our particular choice for the orientations of hypersurface segments came about.

In Fig. 19.a we show a surface tile in 3D, similar to the one that is shown in Fig. 10.d. Its orientation is such that the bounding line element (cf., Fig. 10.a) in the $x|y$-plane at $z = 0$ has a normal, which points towards the enclosed voxel, whereas the bounding line element in the $x|y$-plane at $z = 1$ has a normal, which points away from the enclosed voxel. In analogy to the 3D case (as implemented by VESTA [12]), we show in Fig. 19.b a hypersurface tile that is embedded into 4D (cf., Fig. 12.c). Its orientation is chosen such that the bounding surface element (cf., Fig. 10.c) in the $x|y|z$-subspace at $t = 0$ has a normal, which points towards the enclosed toxel, whereas the bounding surface element in the $x|y|z$-subspace at $t = 1$ has a normal, which points away from the enclosed toxel.
B. R. Schlei, “Hyper-Surface Extraction in Four Dimensions,” Theoretical Division - Self Assessment, Special Feature, a portion of LA-UR-04-2143, Los Alamos (2004) 168.

B. R. Schlei, “A New Computational Framework for 2D Shape-Enclosing Contours,” Image and Vision Computing 27 (2009) 637–647, doi: 10.1016/j.imavis.2008.06.014.

L. Dorst, D. Fontijne, S. Mann, Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry, Morgan Kaufmann, 2007.

C. Perwass, Geometric Algebra with Applications in Engineering, Geometry and Computing, Springer, 2009.

B. R. Schlei, “STEVE - Space-Time-Enclosing Volume Extraction Algorithm, Version 1.0,” Los Alamos Computer Code LA-CC-04-056, Los Alamos National Laboratory.

| Orientation | Path 1 | Path 2 | Path 3 | Path 1 | Path 2 | Path 3 |
|-------------|--------|--------|--------|--------|--------|--------|
| Center ID   | 0      | 32 → 0 → 33 | 33 → 0 → 38 | 38 → 0 → 32 | 33 → 0 → 32 | 32 → 0 → 38 | 38 → 0 → 33 |
|             | 1      | 34 → 1 → 32 | 32 → 1 → 39 | 39 → 1 → 34 | 32 → 1 → 34 | 34 → 1 → 39 | 39 → 1 → 32 |
|             | 2      | 32 → 2 → 35 | 35 → 2 → 40 | 40 → 2 → 32 | 35 → 2 → 32 | 32 → 2 → 40 | 40 → 2 → 35 |
|             | 3      | 36 → 3 → 32 | 32 → 3 → 41 | 41 → 3 → 36 | 32 → 3 → 36 | 36 → 3 → 41 | 41 → 3 → 32 |
|             | 4      | 33 → 4 → 34 | 34 → 4 → 42 | 42 → 4 → 33 | 34 → 4 → 33 | 33 → 4 → 42 | 42 → 4 → 34 |
|             | 5      | 35 → 5 → 33 | 33 → 5 → 43 | 43 → 5 → 35 | 33 → 5 → 35 | 35 → 5 → 43 | 43 → 5 → 33 |
|             | 6      | 34 → 6 → 36 | 36 → 6 → 44 | 44 → 6 → 34 | 36 → 6 → 34 | 34 → 6 → 44 | 44 → 6 → 36 |
|             | 7      | 36 → 7 → 35 | 35 → 7 → 45 | 45 → 7 → 36 | 35 → 7 → 36 | 36 → 7 → 45 | 45 → 7 → 35 |
|             | 8      | 33 → 8 → 37 | 37 → 8 → 46 | 46 → 8 → 33 | 37 → 8 → 33 | 33 → 8 → 46 | 46 → 8 → 37 |
|             | 9      | 37 → 9 → 34 | 34 → 9 → 47 | 47 → 9 → 37 | 34 → 9 → 37 | 37 → 9 → 47 | 47 → 9 → 34 |
|             | 10     | 35 → 10 → 37 | 37 → 10 → 48 | 48 → 10 → 35 | 37 → 10 → 35 | 35 → 10 → 48 | 48 → 10 → 37 |
|             | 11     | 37 → 11 → 36 | 36 → 11 → 49 | 49 → 11 → 36 | 36 → 11 → 36 | 37 → 11 → 49 | 49 → 11 → 36 |
|             | 12     | 39 → 12 → 38 | 38 → 12 → 42 | 42 → 12 → 39 | 38 → 12 → 39 | 39 → 12 → 42 | 42 → 12 → 39 |
|             | 13     | 38 → 13 → 40 | 40 → 13 → 43 | 43 → 13 → 38 | 40 → 13 → 38 | 38 → 13 → 43 | 43 → 13 → 40 |
|             | 14     | 41 → 14 → 39 | 39 → 14 → 44 | 44 → 14 → 41 | 39 → 14 → 41 | 41 → 14 → 44 | 44 → 14 → 41 |
|             | 15     | 40 → 15 → 41 | 41 → 15 → 45 | 45 → 15 → 40 | 41 → 15 → 40 | 40 → 15 → 45 | 45 → 15 → 41 |
|             | 16     | 46 → 16 → 47 | 47 → 16 → 42 | 42 → 16 → 46 | 47 → 16 → 46 | 46 → 16 → 42 | 42 → 16 → 46 |
|             | 17     | 48 → 17 → 46 | 46 → 17 → 43 | 43 → 17 → 48 | 46 → 17 → 48 | 48 → 17 → 43 | 43 → 17 → 48 |
|             | 18     | 47 → 18 → 49 | 49 → 18 → 44 | 44 → 18 → 47 | 49 → 18 → 47 | 47 → 18 → 44 | 44 → 18 → 47 |
|             | 19     | 49 → 19 → 48 | 48 → 19 → 45 | 45 → 19 → 49 | 48 → 19 → 49 | 49 → 19 → 45 | 45 → 19 → 48 |
|             | 20     | 51 → 20 → 50 | 50 → 20 → 38 | 38 → 20 → 51 | 50 → 20 → 51 | 51 → 20 → 38 | 38 → 20 → 50 |
|             | 21     | 50 → 21 → 52 | 52 → 21 → 39 | 39 → 21 → 50 | 52 → 21 → 50 | 50 → 21 → 39 | 39 → 21 → 52 |
|             | 22     | 53 → 22 → 50 | 50 → 22 → 40 | 40 → 22 → 53 | 50 → 22 → 53 | 53 → 22 → 40 | 40 → 22 → 50 |
|             | 23     | 50 → 23 → 54 | 54 → 23 → 41 | 41 → 23 → 50 | 54 → 23 → 50 | 50 → 23 → 41 | 41 → 23 → 54 |
|             | 24     | 52 → 24 → 51 | 51 → 24 → 42 | 42 → 24 → 52 | 51 → 24 → 52 | 52 → 24 → 42 | 42 → 24 → 51 |
|             | 25     | 51 → 25 → 53 | 53 → 25 → 43 | 43 → 25 → 51 | 53 → 25 → 51 | 51 → 25 → 43 | 43 → 25 → 53 |
|             | 26     | 54 → 26 → 52 | 52 → 26 → 44 | 44 → 26 → 54 | 52 → 26 → 54 | 54 → 26 → 44 | 44 → 26 → 52 |
|             | 27     | 53 → 27 → 54 | 54 → 27 → 45 | 45 → 27 → 53 | 54 → 27 → 53 | 53 → 27 → 45 | 45 → 27 → 54 |
|             | 28     | 55 → 28 → 51 | 51 → 28 → 46 | 46 → 28 → 55 | 51 → 28 → 55 | 55 → 28 → 46 | 46 → 28 → 51 |
|             | 29     | 52 → 29 → 55 | 55 → 29 → 47 | 47 → 29 → 52 | 55 → 29 → 52 | 52 → 29 → 47 | 47 → 29 → 55 |
|             | 30     | 55 → 30 → 53 | 53 → 30 → 48 | 48 → 30 → 55 | 53 → 30 → 55 | 55 → 30 → 48 | 48 → 30 → 53 |
|             | 31     | 54 → 31 → 55 | 55 → 31 → 49 | 49 → 31 → 54 | 55 → 31 → 54 | 54 → 31 → 49 | 49 → 31 → 55 |

TABLE I: Triplets of directed paths (cf., Figs 3.e and 3.f) for the octants of the oriented boundary volumes, which have their centers at the predefined locations as depicted in Fig. 4.b. The start and end points of the 192 paths are the potential points of ambiguity, as shown in Fig. 4.c.