Journal of Power Sources 195 (2010) 8168–8176

Contents lists available at ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

High accuracy interface characterization of three phase material systems in three dimensions

P.S. Jørgensen [a,*], K.V. Hansen [a], R. Larsen [b], J.R. Bowen [a]

[a] Fuel Cells and Solid State Chemistry Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark, Frederiksborgvej 399, 4000 Roskilde, Denmark
[b] Department of Informatics and Mathematical Modelling, Technical University of Denmark, Richard Petersens Plads, Building 321, 2800 Lyngby, Denmark

ARTICLE INFO

Article history:
Received 8 December 2009
Received in revised form 10 May 2010
Accepted 23 June 2010
Available online 6 July 2010

Keywords:
Interface characterization
Triple phase boundary
Surface area
Solid oxide fuel cells

ABSTRACT

Quantification of interface properties such as two phase boundary area and triple phase boundary length is important in the characterization of many material microstructures, in particular for solid oxide fuel cell electrodes. Three-dimensional images of these microstructures can be obtained by tomography schemes such as focused ion beam serial sectioning or micro-computed tomography. We present a high accuracy method of calculating two phase surface areas and triple phase length of triple phase systems from sub-voxel accuracy segmentations of constituent phases. The method performs a three phase polygonization of the interface boundaries which results in a non-manifold mesh of connected faces. We show how the triple phase boundaries can be extracted as connected curve loops without branches. The accuracy of the method is analyzed by calculations on geometrical primitives.

1. Introduction

Three-dimensional reconstructions of microstructural data have become an important tool in analysing and gaining understanding of multiphase material systems on the microstructure level [1–3]. These material systems include solid oxide fuel cell (SOFC) electrodes where the distribution and interconnectivity of constituent phases are of critical importance for the electrochemical performance. The volume specific triple phase boundary (TPB) length is one of the most critical performance attributes of SOFCs. The TPB length is the total length of the curves within the microstructure that bound the interface between each of the three constituent phases. The TPBs are important since they are the locations where the electrochemical reactions within the cell take place.

We present here a high accuracy method for quantifying the TPB length and interface area of triple phase systems from a prior partition of the analyzed volume into its constituent phases (henceforth denoted segmentation). This paper can be seen as a direct continuation of our previous work [4] that describes a computational framework for sub-voxel (volume element) accuracy segmentation of microstructural tomography data. The methodology described here enables the extraction of accurate measurements from already obtained segmentations.

TPB length calculations have been carried out before [5,6]. Common to these calculations are that TPB sites are identified by systematically searching through the voxel structure for locations that contain all three types of phases in a small neighbourhood. The result is a grid of TPB sites. The TPB length has then been quantified either by counting the number of TPB sites and weighting them by the voxel dimension or the points are connected and the total length of the line segments is summed. Simply connecting the TPB sites has been reported to cause problems of branching curves [5]. These approaches lead to an overestimation of the TPB curve length because the TPB line segments do not form a smooth curve. A correction factor can be used to account for the overestimation. However, determination of the correction factor is challenging since the factor will be a function of both the sample and the resolution of the voxel grid. Two-dimensional stereological approaches such as [7] essentially fall into the same category since the TPB estimate is based on a count of TPB sites and a correction factor derived from an assumption of microstructure shape.

Ref. [8] employs a method where each phase is expanded slightly and the TPB curves are extracted as the centrelines of the overlapping tube-like volumes. This method will theoretically calculate the TPB length of the segmentation accurately for an infinitesimally small expansion. The segmentation will however need to accurately represent the physical sample interfaces. For a conventional segmentation where voxels are labelled as belonging to one of the constituent phases, the interfaces between phases cannot be represented as smooth surfaces. This means that an infinitesimal

* Corresponding author at: Risø National Laboratory for Sustainable Energy, Technical University of Denmark, Building 778, Frederiksborgvej 399, P.O. Box 49, 4000 Roskilde, Denmark. Tel.: +45 4677 5672; fax: +45 4677 5858.
E-mail addresses: psjq@risoe.dtu.dk (P.S. Jørgensen), karv@risoe.dtu.dk (K.V. Hansen), rl@imm.dtu.dk (R. Larsen), jrbo@risoe.dtu.dk (J.R. Bowen).

0378-7753/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2010.06.083

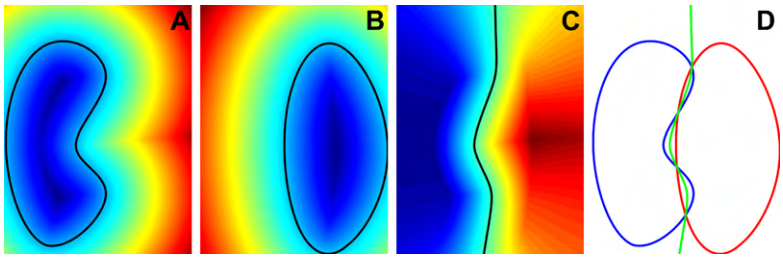

Fig. 1. An example of an implicit three phase segmentation representation. (A–C) The functions ϕ_1, ϕ_2 and ϕ_{12} evaluated on the same domain. Blue shades denote negative values, red shades denote positive values and the black curves show the zero level set of the functions. (D) The zero level sets of the functions ϕ_1, ϕ_2 and ϕ_{12}.

expansion of such a voxel accuracy segmentation will result in line segments that lie exactly on the sides of the voxels. This in turn leads to an overestimation of the TPB curve length as the TPB line segments do not form a smooth curve.

The method presented here uses a different approach to identifying TPB curves. The method reconstructs all the two phase interfaces in the structure as connected polygons. Edges are defined as the line segments of the polygons and vertices are defined as the intersection points between two edges. In this representation of the phase interfaces each polygon shares edges and vertices with adjacent polygons. After the data structure is constructed the TPB curves can be identified as the edges that are shared by three polygons of different phase. This is a conceptually simplified explanation; further details are given in Section 2.5.

In this approach the data structure is in essence the result. The accuracy of the method thus ultimately depends on the accuracy of the interface polygonization. This approach results in TPB curves that all form connected curves without branches. The connectivity of the TPB line segments makes it possible to perform smoothing operations on the TPB line segments to lessen the overestimation problem.

The result of the framework presented in our previous work [4] is phase segmentations represented by the zero iso-level of a signed distance function. This representation of the segmentation provides increased accuracy compared to voxel classification. This is because smooth surfaces can be represented since the interface is not limited to lie exactly halfway between two voxels. The proposed methodology is developed specifically to take advantage of the increased accuracy of sub-voxel segmentations. Conventional voxel accuracy segmentations can however be processed using the same methodology by converting the segmentation to an implicit surface representation.

2. Methodology

An iso-surface polygonization scheme converts a specific iso-level in a volumetric representation into a surface representation defined by a mesh of polygons. Good polygonization techniques such as the Marching Cubes algorithm [9] and its extensions [10] have been around for decades. These techniques are well established and an integral part of many software visualization packages. Common to these polygonization methods is that they search through the voxels in the volume to locate those voxel neighbourhoods that contain values above and below the specified iso-level. Different interpolation schemes are then applied to create polygons with vertices on iso-level locations. However, most of the iso-surface polygonaization techniques are only designed for two phase structures. Polygonization techniques have been developed for the n-phase case [11]. The polygonization methodology presented below is similar to this technique and in many cases simpler. The methodologies do diverge in the problem they attempt to solve; mainly due to the need to collapse two representations of the same interface into one as is described in Section 2.1. Since the polygonization scheme is based primarily on existing techniques the methodology described here will be on a largely conceptual level, hence the reader is referred to the existing literature [9–13] for an in-depth description of the underlying mathematics and algorithms. Focus is placed instead upon the areas of the presented approach that diverge from existing literature.

All figures in this section are constructed as two-dimensional examples for illustration purposes only. Therefore phase boundaries are represented as curves and TPBs as points. However, the algorithm itself works in three dimensions where phase boundaries are calculated as surfaces (polygons) and TPBs as curves (connected polygon edges).

2.1. Segmentation representation

Let us start by introducing the notion of a signed distance function in the context of a three-dimensional surface. The signed distance function of an interface at a location (X, Y, Z) in the segmented volume is defined as the distance to the closest point on the interface, negative on the inside of the surface and positive on the outside of the surface. A more thorough coverage of implicit surfaces and signed distance functions is presented in Ref. [4].

The polygonization method described below takes as input two sub-voxel accuracy segmentations where the interfaces are described implicitly as the zero iso-level of a signed distance function. The concept is exemplified in Fig. 1(A and B). The signed distance function is discretely evaluated on a regular grid of voxels. Such a segmentation can be created as described in Ref. [4]. For a three phase system it is required that a segmentation of two of the constituent phases are provided. Let the three phases of the system be denoted P_1, P_2 and P_3 respectively and let the functions that implicitly contain the interface of two of the phases be denoted ϕ_1 and ϕ_2. A function ϕ_3 is not required since the P_3 interface information can be derived from ϕ_1 and ϕ_2. Fig. 1 shows the segmentation representation. The segmentation representation leads to a double representation of the interface between P_1 and P_2. A polygonal interface representation is sought where each polygon separates two specific phases and where only one polygon represents a specific interface section. A way to collapse the double interface representation to a single interface is thus needed. Since the segmentation of P_1 and P_2 can be performed independently it cannot be assumed that the two representations of the same interface are congruent. The two segmented phases represented by the zero level sets of ϕ_1 and ϕ_2 may overlap in some sections or have small gaps between them in other areas. This means that a definition of when the two interfaces are considered to coincide is needed.

In the locations where the zero level sets of ϕ_1 and ϕ_2 overlap the optimal location of the collapsed surface is halfway in between the two interfaces. This surface can be expressed as the zero level set of the function $\phi_{12} = (\phi_1 - \phi_2)/2$. Recall that ϕ_1 and ϕ_2 are signed distance functions which means that the value at any location corresponds to the distance to the interface. The function ϕ_{12} will thus

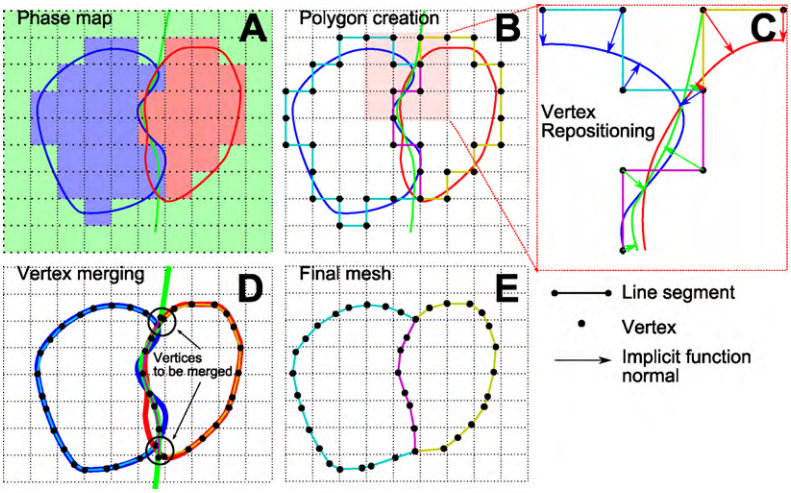

Fig. 2. Two-dimensional illustration of the mesh creation and vertex repositioning.

evaluate to zero at locations that have the same distance to the ϕ_1 and ϕ_2 zero level sets. The division by two ensures that ϕ_{12} is a good approximation to a signed distance function whenever the nearest ϕ_1 and ϕ_2 zero iso-levels are close to parallel.

We define coincidence of the interface between P_1 and P_2 as the locations x on the zero level set of ϕ_{12} for which the distance to both the P_1 and P_2 interface is smaller than a merge threshold $\tau_m(\phi_1(x) < \tau_m \wedge \phi_2(x) < \tau_m \wedge \phi_{12}(x) = 0)$. At these locations the coinciding interfaces will be collapsed to the nearest point on the ϕ_{12} zero level set. The following sections will describe step by step how the interfaces are polygonized and stitched together at the triple phase boundaries and how the zero level set locations are obtained when x no longer represents a continuous location but a discrete voxel location.

2.2. Phase map

A phase map is created from the two implicit functions ϕ_1 and ϕ_2. This phase map corresponds to a conventional voxel accuracy segmentation where each voxel location x has a label $L(x)$ describing what phase it belongs to. This phase map defines the topology of the interface structures. The phase map is constructed at each location x based on a set of basic rules:

(1) $\phi_1(x) \leq 0 \wedge \phi_2(x) > 0$. The location is only inside the ϕ_1 zero level set. Label as phase 1.
(2) $\phi_1(x) > 0 \wedge \phi_2(x) \leq 0$. The location is only inside the ϕ_2 zero level set. Label as phase 2.
(3) $\phi_1(x) \leq 0 \wedge \phi_2(x) \leq 0$. The location is inside both the ϕ_1 and ϕ_2 zero level sets (overlapping interfaces). Label as the phase with the lowest $\phi(x)$ value.
(4) $\phi_1(x) > 0 \wedge \phi_2(x) > 0$. The location is outside both the ϕ_1 and ϕ_2 zero level sets. If $\phi_1(x) > \tau_m \wedge \phi_2(x) > \tau_m$ label as phase 3 otherwise label as the phase with the lowest $\phi(x)$ value.

Rules 1–3 are straight forward and intuitive however the 4th might seem counter intuitive since these locations are clearly outside both the ϕ_1 and ϕ_2 zero level set. The threshold τ_m is used to prevent problems in locations where the ϕ_1 and ϕ_2 zero level sets almost coincide but still have a small gap between them. At these locations a thin layer of phase 3 voxels can form between P_1 and P_2. Note that this threshold typically is set to a very low value depending on the quality of the segmentation, typically between 0 and 1 voxels. A final optional step is to collapse isolated phase regions in the phase map with an unrealistic small volume as these are more likely to be segmentation errors than actual particles. The phase map creation is illustrated in Fig. 2A.

2.3. Voxel accuracy polygonization

The phase map is processed through a very simple initial polygonization scheme. The scheme runs through all voxels in the phase map. Three of the voxels six non-diagonal neighbours are checked and if the neighbour has a different class than the current voxel a quadrilateral is created on the side of the voxel. Only three of the six sides are tested in order to avoid duplicated polygons of the same interface. Quadrilaterals consist of 4 edges that connect pairs of vertices. The concept is illustrated in Fig. 2B. At this initial stage each polygon is unaware of its position in the mesh relative to other polygons and no edges or vertices are shared between polygons. To be able to resolve the connectivity of the polygons a reference to the polygon is inserted into a connectivity matrix. A connectivity matrix exists for each voxel corner in the voxel grid that is used by at least one polygon. The connectivity matrix makes it possible to merge vertices that coincide according to the topology of the phase map.

2.4. Sub-voxel accuracy polygonization

The voxel accuracy polygon structure is ill suited for surface or TPB calculations due to its cubic nature as it overestimates area and length calculations. If no sub-voxel accuracy information was available this would typically be remedied by smoothing the polygon mesh after the connectivity of the polygons had been resolved. Smooth sub-voxel accuracy surface information is however available from the three functions ϕ_1, ϕ_2 and ϕ_{12}. The implicit surfaces described by those functions are supported directly by the intensity information in the raw intensity data [4] and are thus preferred over post-segmentation smoothing.

To obtain polygon vertex coordinates with sub-voxel accuracy a vertex movement scheme is applied to all vertices. The positioning scheme takes advantage of two important properties of signed distance functions:

- The absolute value of the signed distance function evaluated at a certain location corresponds to the distance to the closest point on the zero level set.
- The gradient of a signed distance function evaluated at a certain location corresponds to the direction in which the closest point on the zero level set is found.

These two properties enable us to design a vertex movement scheme that moves each vertex onto the interface described by the zero level set of the signed distance function.

For each vertex position x_v we interpolate the value $\phi(x_v)$ and the gradient $\nabla\phi(x_v)$ of the appropriate signed distance function from the values at the surrounding voxels. The new vertex position is then found at $x_{\text{new}} = x_v - \phi(x_v)\nabla\phi(x_v)$. Given a continuous signed distance function this movement scheme will move the vertex exactly on to the interface represented by the zero level set of the signed distance function. The appropriate function to evaluate is based purely on which set of two phases the polygon that the vertex is part of separates.

- For an interface between P_1 and P_3, ϕ_1 is used.
- For an interface between P_2 and P_3, ϕ_2 is used.
- For an interface between P_1 and P_2, ϕ_{12} is used.

The sub-voxel accuracy repositioning scheme is illustrated in Fig. 2C.

At this point the set of polygons do not constitute surfaces but rather a collection of individual polygons. To form a data structure from which calculations can be extracted the vertices from adjacent polygons must be merged and neighbour connection information must be stored for each polygon. This connectivity information is constructed by processing each connectivity matrix in turn. The connectivity matrix contains a reference to all polygons that according to the topology of the phase map share the same vertex. This vertex is denoted the centre vertex. The index of the polygon in the connectivity matrix corresponds to the polygons relative position to the centre vertex. Using these matrix indices it is possible to efficiently connect all pairs of polygons that share two vertices (an edge). Compared to polygons that separate two phases only, the polygons located at the TPB use three different functions (ϕ_1, ϕ_2 and ϕ_{12}) for vertex repositioning. It is therefore not expected that these vertices are repositioned to the same position. The position of the centre vertex is thus calculated as the average position of the vertices that are to be merged. The vertex merging is illustrated in Fig. 2D and the final mesh can be seen in Fig. 2E.

At the initial polygon creation step a reference to the two voxels that the polygon separates is stored together with the polygon. A powerful dual representation of the phase structure is thus available. A phase map representation that represents the interior of each phase and a polygonal mesh that accurately describes the interface between the phases. The references between polygons and the two voxels they separate facilitate a simple transition between the two representations.

2.5. TPB identification

With the interface topology/morphology of the phases stored in a data structure as described above it is straight forward to locate the edges in the mesh that correspond to a triple phase curve segment. The TPB edges are identified as the outline of the interface surface between phases 1 and 2. In other words, edges that are part of only one polygon that separates a phase 1–2 and both a 1–3 and a 2–3 polygon simultaneously. This definition intuitively provides us with the TPB edges since a TPB must exist where a two phase boundary meets the third phase. This definition also guarantees that the TPB curves will form non-branching curves that either form a loop inside the voxel cube or that intersect the side of the voxel cube.

In practice this scheme is implemented by running through all polygons checking the criteria described above. The qualifying edges are stored in a separate list and linked to their neighbours by identifying when two edges use the same vertex. To obtain accurate TPB measurements it is important that the extracted TPB curves are smooth. However, this will often not be the case. The connectivity information of the edges can however be used to smooth the edges.

A simple smoothing approach has been applied here where each vertex of the edge curve is repositioned by calculating a weighted average of the position of itself and its immediate neighbours. Let x_c, x_p and x_c denoted the current, previous and next vertex positions of a TPB curve loop, respectively. The repositioning step moves the current vertex to the position $x_c = 0.5x_c + 0.25x_p + 0.25x_n$. This smoothing approach can be repeated, each time resulting in a smoother curve of connected edges. However, a higher number of smoothing iterations is not strictly an improvement in accuracy. An increase in smoothing iterations corresponds to an increase in the number of neighbour vertices with non-zero weights. This will result in a shrinkage effect of the TPB curve loops. It is thus necessary to weigh the benefit of the smoothing against the shrinkage effect. A quantitative analysis of this dilemma is available in Section 3.2. Note that the smoothing operations reposition vertices that are still part of the interface mesh. The mesh structure will thus contain the smoothed TPB edges which remain congruent with the surface mesh for visualization purposes.

The finished mesh structure allows very simple calculations of both two phase boundaries and TPBs. Area calculations are performed by simply summing the area of polygons that separate specific phase pairs. In this way the two phase boundary areas between two specific phases can be calculated along with the total interface area of each phase. The TPB length can be calculated by summing the length of the edges that make up the TPB curves. The link between the mesh representation and voxel representation can be used to further characterize the TPB locations. As an example the percolation [6] of each TPB edge can be identified by performing the percolation identification on the voxel grid and then utilizing the reference between the polygons and the voxel grid to check for percolation of the TPB edges.

3. Results

One approach to assessing the accuracy of proposed TPB measurement methods is to compare measured values with physical measurements that are believed to be governed mainly by the TPB length [1]. It is however difficult to determine the cause of the disparities by comparing a TPB length measure based on microstructure calculations to a measure obtained by a model of a directly measurable physical property. The disparity could be due to several factors.

- Errors or insufficient resolution in the segmentation.
- Errors in the computation method used to calculate the TPB length from the segmentation.
- The model that connects the physical measurement to the TPB length could be too simple.

The accuracy analysis presented here takes a different approach. To asses the accuracy of the TPB length and surface area measurement method alone, tests are constructed that seek to eliminate segmentation errors and model errors. This is achieved by applying the method to spheres of varying diameter in a set-up similar to Ref. [8]. The advantage of using spheres for the analysis is 3-fold.

(1) Segmentation error can be eliminated since a sphere can be constructed as an implicit surface of a signed distance function without errors. This is achieved by calculating the exact Euclidian distance to the centre of the sphere at each voxel location. To obtain a sphere with a specific radius the radius is simply subtracted from all voxels. This representation is error free in the sense that it is the most accurate signed dis-

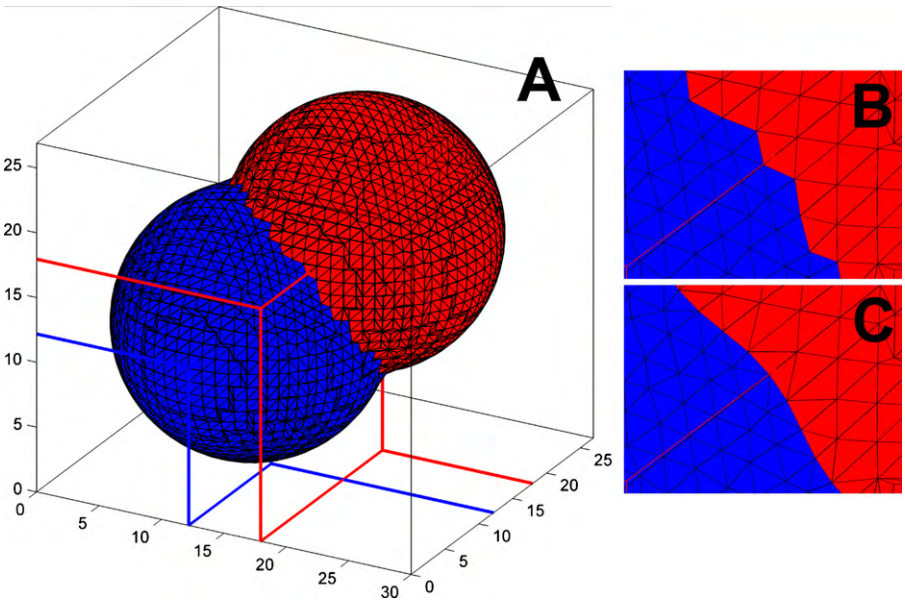

Fig. 3. The intersecting sphere test set-up. (A) An example of the mesh reconstruction without smoothing. (B) Enlargement of a TPB section of A. (C) The same enlargement with two smoothing iterations applied.

tance function representation that can be achieved at a given resolution.

(2) The surface area of a sphere is defined as $A_s = 4\pi r$, where r is the radius of the sphere. Since this is an exact formula and not a model the calculations of surface area can be directly compared. For the TPB length calculations a similar exact formula can be obtained for the intersection of two spheres of radius r, at the distance r from each other. The length of the intersection curve between the two spheres, and thus the TPB length, is defined as $L_{\text{int}} = 2\pi\sqrt{r^2 - (r/2)^2}$.

(3) The segmentation sampling resolution error can be assessed by varying the resolution of the sphere representation on the voxel grid. Let the radius of the sphere be measured in voxel widths. Increasing the radius will then result in a subsequent increase in sampling resolution. Recall that the basis of the surface area and TPB length calculations is a voxel grid with finite resolution.

To show the accuracy of the proposed method on voxel accuracy segmentation the following changes were made to the implicit surfaces used as input for the method. The signed distance map was first constructed as outlined above for the implicit surface representation of the spheres. The voxel grid was then discretized into three classes based on the voxel value of both implicit functions using the same methodology as described in Section 2.2.This segmentation represents the best achievable voxel accuracy segmentation of the sphere. The voxel accuracy segmentation was then converted back to a set of signed distance function implicit surfaces but loosing the advantage of the sub-voxel accuracy in the process. Both a sub-voxel accuracy and voxel accuracy set-up were used in the tests. All tests were performed with radii ranging from 1 to 30 voxels with increments of 0.25 voxel.

Two different set-ups where used for the sphere centres, an aligned set-up and a skewed set-up:

- The aligned set-up sets the centre of the spheres in the centre of a voxel. This set-up creates a set-up with many symmetric axes that coincide with the three major directions of the voxel grid. The TPB length calculations place the second sphere the distance r away in the direction of the vector $V_{\text{align}} = [1\,0\,0]^T$. This set-up represents a best case scenario where the most accurate reconstruction result can be expected.
- The skewed set-up represents a more challenging case where the centres of the spheres are not in the centre of voxels. The offset from the centre of the voxel was chosen to be 0.1817. This offset was selected to not be a simple fraction between 0 and 1 but the value could have been any other number. The TPB length calculations place the second sphere the distance r away from the first in the direction of the vector $V_{\text{skew}} = [1\,1\,1]^T$. This set-up represents a worst case scenario where the least accurate results can be expected. The set-up can be seen in Fig. 3.

It is reasonable to assume that the accuracy of a real microstructure with a similar radius of curvature of its interface boundaries would be somewhere between the accuracy of these worst case and best case set-ups. In total four set-ups were tested: Aligned spheres and skewed spheres both with and without sub-voxel accuracy. The sphere results are reported as relative errors calculated as $\text{err} = (m_{\text{exact}} - m_{\text{measured}})/m_{\text{exact}}$, where m_{exact} is the exact analytical quantity.

3.1. Surface area

The surface area of the sphere in each of the 4 set-ups was calculated as described in Section 2.5. For the surface area calculations only one sphere was used in each of the set-ups. The results of the surface area calculations can be seen in Fig. 4. The general trend for all 4 set-ups is the same. The error is large for spheres with small radii and has an asymptotic convergence as the radii increases. To increase the readability of the figures they only show the relative error for radii larger than 3.

As can be seen the results for the aligned and skewed set-up are almost identical in the case of sub-voxel accuracy. This is because the signed distance function evaluated on the voxel grid can represent the skewed case just as well as the aligned case. The voxel accuracy tests perform worse than the sub-voxel accuracy tests for large radii. This is because the discrete voxel classification cannot accurately represent the sphere; the surface of the polygonized sphere will thus have artefacts caused by the voxelated surface structure of the voxel accuracy representation. The noise in the

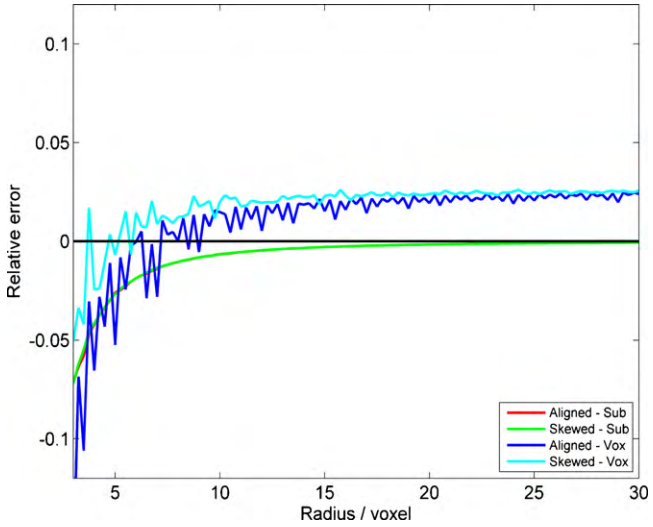

Fig. 4. Surface area results. The relative error of the surface area calculations as a function of sphere radius for the 4 different set-ups. Note that results of the aligned and skewed sub-voxel precision set-ups are very similar and thus indistinguishable from each other in the figure.

curves is caused by some radii resulting in surfaces that are more aligned with the voxel grid than others.

By observing Fig. 4 the consequence of two kinds of errors can be observed. The two types of error are illustrated in Fig. 5.

(1) The first error is sampling frequency. The sphere is represented by a finite number of polygons and the surface area is calculated as the combined area of these. The total measured area will thus always be smaller than the theoretical value assuming that the polygon vertices are placed exactly on the surface of the sphere. This error is clearly visible in Fig. 4 where all the tests have a large under estimation for small radii and a decreasing under estimation for larger radii. The limit of the sampling frequency error is zero as the radius approaches infinity.
(2) The second error is sampling accuracy. This is an error in the location of the vertices of the polygons. An extreme case would be to use the sides of the voxel segmentation as polygons directly. This scheme would cause a severe over estimation of the surface area. This error can be made very low but it will always be present to some degree and the error will always be an overestimation assuming that the sampling accuracy error has no preferred direction.

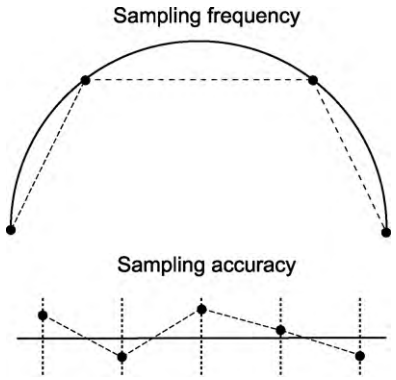

Fig. 5. The two types of errors in the reconstruction. Sampling frequency error is the error introduced by approximating smooth curves with line segments. Sampling accuracy error is the error introduced by inaccurate vertex positioning.

Any measurement will thus always be the result of two errors that influence the measurement in different directions. The surface area measurements are heavily influenced by sampling frequency error for small radii thus giving a large under estimation. The sub-voxel accuracy measurements generally have a very small sampling accuracy error. This can be seen by the fact that the underestimation decreases towards zero as the radii becomes larger. If a significant sampling accuracy error was present we would expect the estimation error to cross zero at some point. The voxel accuracy tests in Fig. 4 both cross zero, indicating that they are affected by sampling accuracy error. This also explains why the error does not tend to zero for large radii.

The sub-voxel test becomes very accurate for large radii with a relative error of −0.17% for a radius of 20, where the aligned and skewed voxel accuracy tests have a relative error of approximately 2.0% and 2.5% respectively for the same radius.

The radius of curvature of interfaces in a real sample will be a distribution of values rather than a single value. These interfaces will have a random alignment with respect to the three major axes of the voxel grid. The relative error of a random interface orientation is expected to be bounded by the aligned and skewed set-ups since they correspond to a worst case and best case scenario. For this purpose sub-voxel and voxel precision results should be treated separately. In general the plots of relative error versus sphere radius can be used in two ways.

- To asses the accuracy and precision of a calculation on a physical microstructure. The worst case accuracy can be assessed as the largest absolute relative error of the aligned and skewed set-up results. Only the values in the expected radius of curvature range of the sample should be used. The precision can be assessed based on the difference between the signed highest and lowest value in the same range of the aligned and skewed set-up. A high precision is characterized by relative errors that are close to constant (a smooth flat relative error curve in Fig. 4) and a small difference between the relative errors of the aligned and skewed set-ups. In such a case there is a systematic under or over estimation that can be systematically corrected in the interface calculations.
- To choose image resolution and slice frequency before data acquisition. Based on the assessed curvature of radius of the sample the expected accuracy of the results can be assessed before data requisition begins. As an example supposed an estimate of the surface area of interfaces in a sample is wanted with maximum ±1.5% error. From inspection of two-dimensional micrographs the typical radius of curvature of the grains is assessed to range between 200 and 450 nm. From Fig. 4 it is observed that a radius of curvature of between 4 and 9 voxels has relative errors between −4% and −1% for a sub-voxel precision segmentation. Estimates in this range can thus be systematically corrected for the 2.5% underestimation we would expect when centring the range on zero. The maximum error with this correction in place would be ±1.5%. The maximum voxel size that would ensure this accuracy is 50 nm (200 nm/4 voxels). Note that this maximum error is only a measure of the error introduced by the sampling resolution and the calculation method. Errors introduced by inaccurate segmentation need to be assessed separately.

3.2. TPB length

Fig. 6 shows the TPB length calculations with zero and two smoothing iterations for the four different set-ups. Note that for the TPB figures, the relative errors are plotted versus the radius of the TPB circle at the intersection rather than the radius of the spheres. This is done to be able to asses the accuracy versus the radius of curvature of the TPB curve loop.

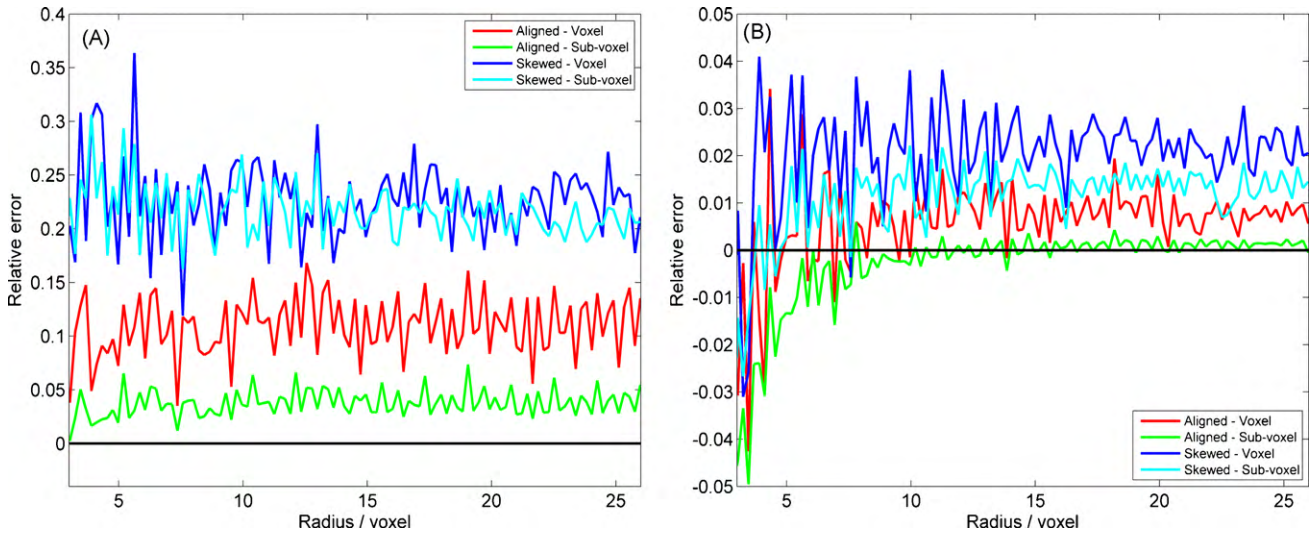

Fig. 6. TPB length accuracy for the 4 set-ups for no smoothing and two smoothing iterations. (A) The relative error of the TPB length calculations without smoothing. (B) The relative error of the TPB length calculations with 2 smoothing iterations.

Fig. 6A shows the accuracy for the 4 set-ups with no smoothing. For TPB curve radii larger than 5 voxels an overestimation of approximaetly 5–25% of the TPB length is observed for the 4 set-ups. Note that there is a large difference between the results for the aligned and skewed set-ups and each curve has significant noise. This indicates that the calculated result is very dependent on the orientation of the surface relative to the voxel grid. As a result, calculations performed without any smoothing iterations would have a large confidence interval.

The overestimation is more severe for the skewed set-up and for voxel precision surface representations. This is as expected since the reconstructed TPB curve loop of the skewed set-up by construction is not parallel with the major axes of the voxel grid tessellation, thus resulting in a rough TPB curve loop. A visualization of the rough reconstruction of the TPB curve can be seen in Fig. 3. Clear artefacts from the original voxel structure can be seen in the curve. Even though these artefacts are much smaller than if the vertex positioning scheme had not been applied they still cause a substantial lengthening of the TPB curve.

If the shape of the phase structures where known it would be possible to obtain a much better initial TPB curve reconstruction. The algorithm described here however works on any phase structure. The algorithm thus reconstructs the spheres as it would any other structure.

Fig. 6B shows the accuracy for the four set-ups with two smoothing iterations applied. The two smoothing iterations increase the accuracy across the 4 set-ups compared to not applying smoothing. For radii larger than 5 voxels a relative error of between −1.1% and 3.8% is observed for the voxel accuracy test and a relative error of between −1.6% and 2.2% is observed for the sub-voxel accuracy tests. The figures can be used to give a rough confidence interval of the TPB estimates analogous to the discussion in Section 3.1. Similarly the figures can be used to select the required voxel size for a given TPB estimation accuracy. The typical radius of curvature range of the TPB curves is not easily extracted from two-dimensional slices and a three-dimensional reconstruction of a similar sample is thus likely required to select the required voxel size of future samples.

The relative errors of the 4 set-ups appear to reach an asymptotic value with added noise for radii larger than 5 in Fig. 6A and for radii larger than 10 in Fig. 6B. These radii indicate the minimum radius of curvature of the TPB curve that is sufficient to obtain the maximum accuracy of the reconstruction algorithm using the given smoothing scheme.

Fig. 6A and B shows how applying a few iterations of smoothing dramatically reduces the relative error of the reconstruction for radii larger than 5 voxels. The number of smoothing iterations is a parameter of the TPB curve reconstruction and can in theory be set at arbitrarily large values. Increasing the number of smoothing iterations can be seen as including a larger neighbourhood of vertices in the weighted sum calculation of the new vertex position (Section 2.5). Since the TPB curves form loops this means that for the number of iterations approaching infinity the TPB curves will collapse to a single point that is the centre of gravity of the loop. In practice the amount of smoothing iterations will have to be chosen based on the radius of curvature of the TPB curves. Fig. 7 shows the skewed sub-voxel accuracy set-up for smoothing iterations between zero and ten. It is seen that a high number of smoothing operations result in slower convergence but increases the asymptotic accuracy and precision. For 10 smoothing iterations the calculated TPB length estimate is within ±0.4% of the theoretical value but a radius of curvature of 15 or higher is required to obtain this accuracy. For

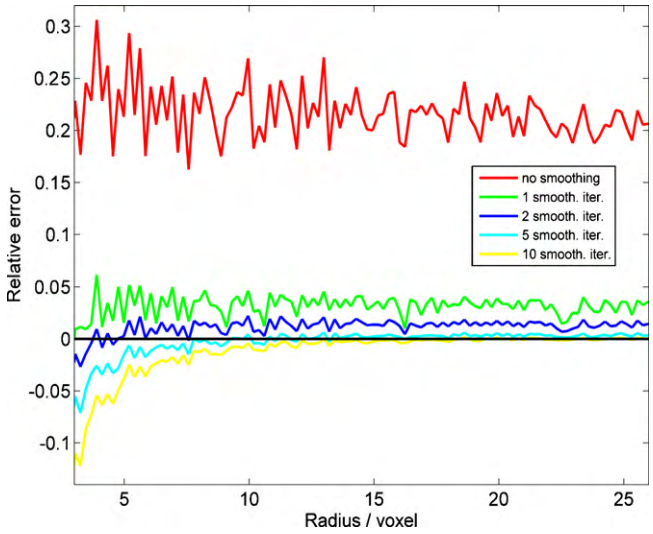

Fig. 7. The effect of smoothing. The relative error of the TPB length calculations for varying amounts of smoothing iterations on the skewed sub-voxel set-up.

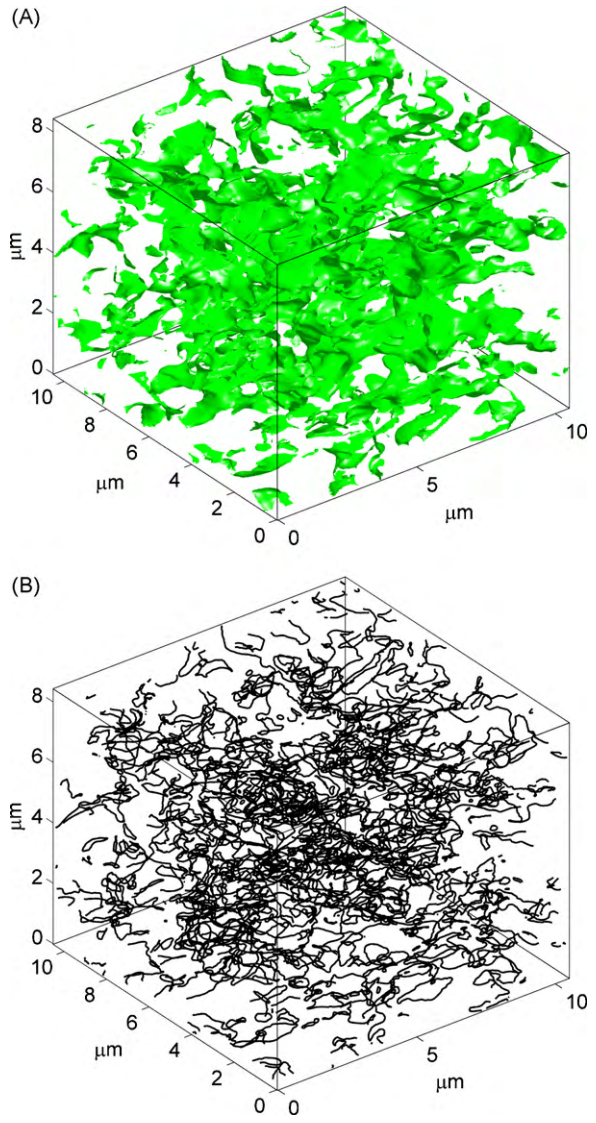

Fig. 8. LSC/CGO electrode interface visualization. (A) The pore/LSC interface. (B) The TPB curves.

smaller radii of curvature 10 smoothing iterations can result in a severe under estimation of the TPB length. This shows the significance of choosing a smoothing scheme that is appropriate for the chosen voxel dimension and sample structure size. Fig. 7 also gives a hint of the accuracy that can be achieved as tomography resolution and quality improves.

3.3. Interface characterization of a SOFC electrode

The proposed characterization calculations were applied to a segmentation of a real SOFC sample. The sample is a composite SOFC cathode consisting of 50/50 wt.% strontium-substituted lanthanum cobaltite (LSC) and gadolinia-substituted ceria (CGO), similar to Ref. [14]. The sub-voxel segmentation of this sample has previously been described in Ref. [4] along with figures showing the raw electron microscopy image data and the segmentation results. The interface calculations were performed on a grid of $180 \times 180 \times 190$ voxels with a voxel dimension of 58.6 nm × 58.6 nm × 44.5 nm. The collapse range was set to 0.5 voxels and connected material grains smaller than 10 voxels were collapsed before the polygonization. These two parameters were described in Section 2.2. Two smoothing iterations were applied to the TPB curves. Fig. 8A shows a visualization of the pore/LSC two phase boundaries and Fig. 8B shows the extracted TPB curves. Table 1 shows the calculated interface measurements for the sample. The percolating TPB length was calculated by only summing TPB line segments that have a percolating path in each phase to their respective source side: the LSC phase to the side facing the electrolyte and the CGO and pore phases to the opposite side.

Table 1
Interface measurements on a LSC/CGO cathode. All measurements are volume specific.

L_V (TPB)	1.808 μm^{-2}
L_V (percolating TPB)	1.413 μm^{-2}
S_V (pore surface area)	1.074 μm^{-1}
S_V (CGO surface area)	1.896 μm^{-1}
S_V (LSC surface area)	1.822 μm^{-1}
S_V (pore/CGO surface area)	0.575 μm^{-1}
S_V (pore/LSC surface area)	0.500 μm^{-1}
S_V (CGO/LSC surface area)	1.322 μm^{-1}

As discussed at the beginning of Section 3 the accuracy of the calculated interface parameters on real samples are influenced by several factors. Thus, this investigation of a real SOFC electrode is provided as an example of the applicability of the methodology to real data and not as the validation of the methodology.

3.4. Computational cost

The described method was implemented by the authors in the C++ programming language. This implementation was used for both the analysis on geometrical primitives and the real electrode. The values presented in Table 1 including the percolation analysis took 107 s to calculate on a laptop with an Intel® Core™ 2 T7200 2 GHz CPU and 2 GB of memory. The analyzed already segmented data consisted of 6.1 million voxels. Of the 107 s, 18 s were taken up by the polygonization resulting in a mesh of 2.2 million triangles.

3.5. Future work

The algorithm as described above has a direct correspondence between the resolution of the voxel grid and the resolution of the constructed polygon mesh. Many algorithms exist for polygonal subdivision [15] such as Catmull–Clark subdivision [16]. The resulting increase in polygons could increase the accuracy of both the area and TPB estimates by reducing the influence of sampling frequency error without increasing voxel resolution.

More advanced smoothing techniques could be employed for the TPB vertex smoothing, for instance by fitting splines to the TPB vertex positions. The use of splines to represent the TPB curves would reduce the error introduced by sampling frequency since the spline representation can be evaluated continuously.

4. Conclusion

In this paper a method for accurate calculation of two phase interface area and TPB curve length is presented. The method reconstructs the phase interfaces as a non-manifold polygonal mesh of the structure based on sub-voxel accuracy segmentations of two of the phases. After the creation of the mesh the two phase interface areas are extracted as the sum of the area of polygons that separate sets of two phases. The TPB curve is extracted as the line segments in the mesh that join three polygons of different phase. This TPB curve definition ensures that the extracted TPB curve is a single connected loop of line segments without branches.

The accuracy of the method is analyzed based on calculations performed on perfect spheres of varying radius. The accuracy of the method is dependent on the radius of curvature of the phase

structures in the sample. For radii of curvature larger than 5 voxels a relative error between −1.6% and 2.2% was achieved as a worst case estimate for TPB calculations on sub-voxel accuracy setups with two smoothing iterations. The TPB estimate is shown to have asymptotical very high accuracy for future increases in voxel resolution.

It is shown how graphs of the relative error as a function of radius can be utilized to assess the accuracy of the interface calculations based on knowledge of typical radii of curvature of the interfaces and the TPBs.

The reconstruction algorithm is used to reconstruct the phase interfaces of a SOFC electrode. The two phase interface areas and TPBs are calculated and a qualitative overview of the extracted TPB curves is presented.

Acknowledgements

This work was supported financially by The Programme Commission on Sustainable Energy and Environment, The Danish Council for Strategic Research, via the Strategic Electrochemistry Research Center (SERC) (www.serc.dk), contract no. 2104-06-0011.

References

[1] J.R. Wilson, W. Kobsiriphat, R. Mendoza, H. Chen, J.M. Hiller, D.J. Miller, K. Thornton, P.W. Voorhees, S.B. Adler, S.A. Barnett, Nat. Mater. 5 (2006) 541–544.
[2] F. Lasagni, A. Lasagni, M. Engstler, H.P. Degischer, F. Mucklich, Adv. Eng. Mater. 10 (2008) 62–66.
[3] D. Gostovic, J.R. Smith, D.P. Kundinger, K.S. Jones, E.D. Wachsman, Electrochem. Solid-State Lett. 10 (2007) B214–B217.
[4] P.S. Jørgensen, K.V. Hansen, R. Larsen, J.R. Bowen, Ultramicroscopy 110 (2010) 216–228.
[5] J. Golbert, C.S. Adjiman, N.P. Brandon, Ind. Eng. Chem. Res. 47 (2008) 7693–7699.
[6] J.R. Wilson, M. Gameiro, K. Mischaikow, W. Kalies, P.W. Voorhees, S.A. Barnett, Microsc. Microanal. 15 (2009) 71–77.
[7] S. Zhang, M. Lynch, A.M. Gokhale, M. Liu, J. Power Sources 192 (2009) 367–371.
[8] H. Iwai, N. Shikazono, T. Matsui, H. Teshima, M. Kishimoto, R. Kishida, D. Hayashi, K. Matsuzaki, D. Kanno, M. Saito, H. Muroyama, K. Eguchi, N. Kasagi, H. Yoshida, J. Power Sources 195 (2010) 955–961.
[9] W.E. Lorensen, H.E. Cline, SIGGRAPH 87, in: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, vol. 1, 1987, pp. 163–169.
[10] T.S. Newman, H. Yi, Comput. Graph. 30 (2006) 854–879.
[11] C.C.L. Wang, Comput. Aided Des. 39 (2007) 35–50.
[12] J. Bloomenthal, B. Wyvill, Introduction to Implicit Surfaces, first ed., Morgan Kaufmann Publishers Inc., San Francisco, 1997, p. 332.
[13] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, first ed., Springer, New York, 2003, p. 296.
[14] F. Tietz, A. Mai, D. Stöver, Solid State Ionics 179 (2008) 1509–1515.
[15] J. Warren, H. Weimer, Subdivision Methods for Geometric Design, first ed., Morgan Kaufmann, San Diego, 2001, p. 320.
[16] E. Catmull, J. Clark, Comput. Aided Des. 10 (1978) 350–355.