

UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie chimique et biotechnologique

DÉVELOPPEMENT D'UNE MÉTHODE DE MESURE
DE LA LONGUEUR DE LA COUCHE LIMITE
TRIPHASÉE UTILISANT UNE MÉTHODE
TOMOGRAPHIQUE

Thèse de doctorat
Spécialité : génie chimique

Tony RIVARD

Jury: François GITZHOFER (co-directeur)
Nicolas ABATZOGLOU (co-directeur)
Gessie BRISARD
Ryan Gosselin (Rapporteur)
Oumarou SAVADOGO

À ma femme, ma chère Kaki
À mes enfants, Nicolas, Jérémie et Zachary
Et à mon frère Luc qui du haut du ciel m'a encouragé à terminer

RÉSUMÉ

Dans les piles à combustible à électrolyte solide, l'interface électrode/électrolyte appelée la couche limite triphasée est le site de la réaction électrochimique permettant aux piles à combustible de fournir de la puissance. Le présent travail propose une approche tomographique par faisceau d'ions focalisés pour mesurer la longueur de la couche limite triphasée. Les images sont ensuite capturées par microscopie à balayage électronique et sont transformées en matrices binaires pour le traitement Matlab. Un algorithme, basé sur la théorie de Hoshen-Kopelman, a ensuite été appliqué pour mesurer la longueur de la couche limite triphasée et le pourcentage de connectivité. Cet algorithme a été appliqué sur un empilement de 80 images obtenues par tomographie d'une cathode réelle obtenu par sérigraphie. Le résultat obtenu est de 0,0142 voxels/voxels³, ce qui équivaut à 207,2 nm/nm³ selon le calcul en fonction de la longueur caractéristique d'un voxel. Une courbe de calibration a été produite afin de transformer la valeur de la longueur de la couche limite triphasée en valeur de résistance au transfert de charges par l'entremise du spectre d'impédance, mesurée à partir d'un système de microélectrodes à motifs.

Mots-clés : Tomographie par faisceau d'ions focalisés, Coupes successives, Piles à électrolyte solide, Couche limite triphasée, Électrochimie.

REMERCIEMENTS

Dans un premier temps, je tiens à remercier le Pr. François Gitzhofer et le Pr. Nicolas Abatzoglou qui m'ont supervisé tout le long de ce travail. Ce projet de recherche m'a permis de me dépasser en raison des difficultés techniques rencontrées et je n'y serais jamais arrivé sans leur support. Je voudrais également remercier les gens du CRN2 pour leur soutien et pour leur précieuse collaboration, notamment, M. Jean Beerens et M. Étienne Grondin pour l'aide apportée pour la tomographie. J'aimerais aussi souligner l'apport de M. Pierre Langlois pour son aide à la photolithographie. Je voudrais remercier particulièrement M. Gilles Grondin du département de biologie pour l'aide importante pour l'imprégnation des échantillons. Il importe de souligner la contribution de M. Steve Allen pour son aide à la traduction du code et son support pour l'utilisation du serveur Mammouth. Je porte toute ma gratitude aux divers techniciens en mécanique, M. Patrice Poulin, M. Francis Barette, et M. Henri Gauvin pour leur précieuse aide technique. Aussi, un gros merci aux professionnels du Centre de Caractérisation des Matériaux, en particulier M. Stéphane Gutierrez et Mme Irène Kelsey, pour leur obligeante collaboration. Je salue tout le personnel administratif pour leur disponibilité et leur serviabilité. Je tiens à remercier les organismes subventionnaires SOFC Canada Network et le CRSNG pour leur support financier. Finalement, j'aimerais remercier les membres de ma famille et de ma belle-famille, particulièrement mon beau-père, M. Réal Montambeault, qui m'ont beaucoup encouragé lors des moments difficiles rencontrés lors de ce doctorat.

Un grand merci, enfin, à ceux que j'aurai omis de nommer par inadvertance et qui, de près ou de loin, ont été impliqués dans ce projet de recherche.

TABLE DES MATIÈRES

RÉSUMÉ.....	i
REMERCIEMENTS.....	iii
TABLE DES MATIÈRES.....	1
LISTE DES FIGURES.....	3
LISTE DES TABLEAUX.....	4
LEXIQUE.....	5
LISTE DES ACRONYMES.....	6
CHAPITRE 1 – INTRODUCTION.....	7
SECTION 1.1 – GÉNÉRALITÉS.....	7
SECTION 1.2 – OBJECTIFS DU PROJET.....	9
SECTION 1.3 – HYPOTHÈSE DE RECHERCHE.....	10
CHAPITRE 2 – RECHERCHE BIBLIOGRAPHIQUE.....	12
SECTION 2.1 – TECHNIQUE DE COUPES SUCCESSIVES.....	12
SECTION 2.2 – PHOTOLITHOGRAPHIE D’ÉLECTRODES À MOTIFS.....	15
SECTION 2.3 – TECHNIQUE GLYCINE-NITRATE.....	18
SECTION 2.4 – AUTRES ARTICLES IMPORTANTS.....	19
CHAPITRE 3 – DESCRIPTION DES TECHNIQUES EXPÉRIMENTALES.....	21
SECTION 3.1 – SYNTHÈSE PAR MÉTHODE GLYCINE-NITRATE.....	21
SECTION 3.2 – SÉRIGRAPHIE DES ÉCHANTILLONS DE CATHODE.....	23
SECTION 3.3 – MÉTHODE D’ENROBAGE DE L’ÉCHANTILLON.....	24
SECTION 3.4 – ANALYSE TOMOGRAPHIQUE.....	26
SECTION 3.5 – PROBLÈMES LIÉS À L’INTERPRÉTATION DES COUPES TOMOGRAPHIQUES.....	30
SECTION 3.5.1 – PROBLÈME DE CORRESPONDANCE.....	31
SECTION 3.5.2 – PROBLÈME DE PIVOT.....	31
SECTION 3.5.3 – PROBLÈME DE BRANCHEMENT.....	32
SECTION 3.5.4 – PROBLÈME D’EFFET RIDEAU.....	33
SECTION 3.6 – PHOTOLITHOGRAPHIE.....	34
SECTION 3.7 – PUVÉRISATION CATHODIQUE MAGNÉTRON.....	35
SECTION 3.8 – SPECTROSCOPIE D’IMPÉDANCE ÉLECTROCHIMIQUE.....	36
SECTION 3.9 – ANALYSE PAR ÉLECTRODE DE CONTACT.....	39
CHAPITRE 4 – ANALYSE NUMÉRIQUE ET RÉSULTATS EXPÉRIMENTAUX.....	40
SECTION 4.1 – TRANSFORMATION DES IMAGES EN MATRICES.....	40
SECTION 4.2 – GÉNÉRATION DES MATRICES TRIDIMENSIONNELLES.....	40
SECTION 4.3 – DESCRIPTION DU FILTRE NUMÉRIQUE.....	42
SECTION 4.4 – MESURE DE LA LONGUEUR DE LA COUCHE LIMITE TRIPHASÉE.....	43
SECTION 4.5 – CALCUL DE LA LONGUEUR CARACTÉRISTIQUE D’UN VOXEL.....	43
SECTION 4.6 – ALGORITHME DE HOSHEN-KOPELMAN.....	48
SECTION 4.7 – ANALYSE DE SENSIBILITÉ.....	49
SECTION 4.8 – COURBE DE CALIBRATION.....	51

CHAPITRE 5 – CONCLUSION.....	55
SECTION 5.1 – SOMMAIRE.....	55
SECTION 5.2 – CONTRIBUTIONS.....	57
SECTION 5.3 – TRAVAUX FUTURS.....	57
LISTE DES RÉFÉRENCES.....	59
ANNEXE A – ARTICLE DU CANADIAN JOURNAL OF CHEMICAL ENGINEERING ..	62
ANNEXE B – GRAPHE LOGIQUE DES PRINCIPAUX SOUS-PROGRAMMES	70
ANNEXE C – ALGORITHME DU PROGRAMME MATLAB	72

LISTE DES FIGURES

1.1	Schéma de principe d'une pile à combustible SOFC	7
1.2	Schéma des principaux mécanismes de diffusion à travers l'électrolyte	8
1.3	Distinction entre les sites TPB connectés et les sites non-connectés	9
2.1	Géométrie anodique parallèle	16
2.2	Microscopie à transmission de poudre obtenus par GNP	19
3.1	Vue de face de la machine GNP	21
3.2	Vue de profil de la machine GNP	21
3.3	Photo MEB de la poudre GNP (10k X)	23
3.4	Photo MEB de la poudre GNP (1.5k X)	23
3.5	Photo MEB de la cathode déposée par sérigraphie (18k X)	24
3.6	Opacité de l'échantillon en fonction de la concentration de cérium ajouté	25
3.7	Photo de l'appareillage FIB	27
3.8	Photo d'une image tomographique d'un échantillon de cathode (12k X)	27
3.9	Exemple d'une échantillon d'image tomographiable et non tomographiable	28
3.10	Schéma du système de coupe et de capture d'images	29
3.11	Schéma de principe de la transformation des images tomographiques en matrices binaires	30
3.12	Illustration du problème de correspondance	31
3.13	Photo d'une coupe illustrant le problème de pivot	32
3.14	Illustration du problème de branchement	33
3.15	Photo d'une coupe illustrant le problème d'effet rideau	33
3.16	Illustration qui définit les étapes d'une photolithographie	34
3.17	Masque utilisé pour la fabrication de la courbe de calibration	35
3.18	Schéma de principe de la pulvérisation cathodique magnétron	36
3.19	Modèles de circuit équivalent utilisés en spectroscopie d'impédance	37
3.20	Exemple de spectre d'impédance	38
3.21	Modèle de circuit équivalent de Voigt simplifié	38
3.22	Schéma de principe de l'analyseur de microélectrode à motif	39
4.1	Illustration représentant l'expansion matricielle selon les faces	41
4.2	Illustration représentant l'expansion matricielle selon les faces et les arêtes	41
4.3	Représentation des voxels avant le filtre	43
4.4	Représentation des voxels après le filtre	43
4.5	Schéma des 5 cas d'étude de la longueur d'un voxel	45
4.6	Schéma des 3 géométries utilisées pour l'étude de la directionnalité	46
4.7	Graphique des écarts moyens en fonction de la géométrie et du cas étudié	47
4.8	Schéma de principe des 3 étapes de l'algorithme de Hoshen-Kopelman 3D	48
4.9	Courbe de sensibilité de la longueur de la TPB en fonction du nombre d'images retirées	50
4.10	Photo MEB de la cathode à motifs pour la fabrication de la courbe de calibration	52
4.11	Spectroscopie d'impédance des électrodes à motifs	53
4.12	Courbe de calibration de la résistance au transfert de charge en fonction du périmètre	53

LISTE DES TABLEAUX

3.1	Paramètres utilisés pour la technique GNP.....	22
4.1	Tableau des résultats de directionnalité et calcul d'erreur.....	47
4.2	Résultats de l'analyse de sensibilité.....	50

LEXIQUE

Terme technique	Définition
Pixel	Le pixel représente le plus petit élément constitutif d'une image numérique (PICTure ELement)
Voxel	Le voxel est l'équivalent 3D du pixel (VOLumetric piXEL)

LISTE DES ACRONYMES

Acronyme	Définition
2D	Deux Dimensions
3D	Trois Dimensions
EXP_GDC	Matrice d'expansion du matériau électrolytique
EXP_LSM	Matrice d'expansion du matériau cathodique
FIB	Faisceau d'ions focalisé
GDC	Cérium Dopé au Gadolinium
GNP	Procédé Glycine Nitrate
LSM	Lanthane Strontium Manganate
MAT_GDC	Matrice représentant le matériau électrolytique
MAT_LSM	Matrice représentant le matériau cathodique
MAT_TPB	Matrice représentant la couche limite triphasée
MEB	Microscope Électronique à Balayage
MIP	Porosimétrie par Intrusion de Mercure
O/C	Ratio atomique d'Oxydant sur Carburant
SDC	Cérium Dopé au Samarium
SOFC	Pile à Combustible à Oxyde Solide
TPB	Couche Limite Triphasée
UV	Ultraviolet
YSZ	Zirconium Stabilisé à l'Yttrium

CHAPITRE 1

Introduction

1.1 Généralités

Depuis quelques années, de nouvelles connaissances techniques et scientifiques ont conduit à de nouvelles technologies qui ont un important potentiel d'innovation. Citons, comme exemples, la technologie des membranes industrielles ou encore le recours au plasma pour faire des films extrêmement minces. La fabrication de piles à combustible à électrolyte solide (SOFC) nécessite une série de couches de matériaux composites formant l'anode, l'électrolyte et la cathode. La technologie de dépôt par plasma thermique fut utilisée pour construire un tel arrangement de matériaux composites complexe qui convertit directement l'énergie chimique en énergie électrique par le biais de réactions électrochimiques. La température optimale d'opération d'une pile à combustible à électrolyte solide dépend de ses matériaux de fabrication. La figure 1.1 illustre le fonctionnement d'une pile à combustible de type SOFC. La description des composantes de la pile à combustible étudiée dans cette thèse est écrite au chapitre 3.

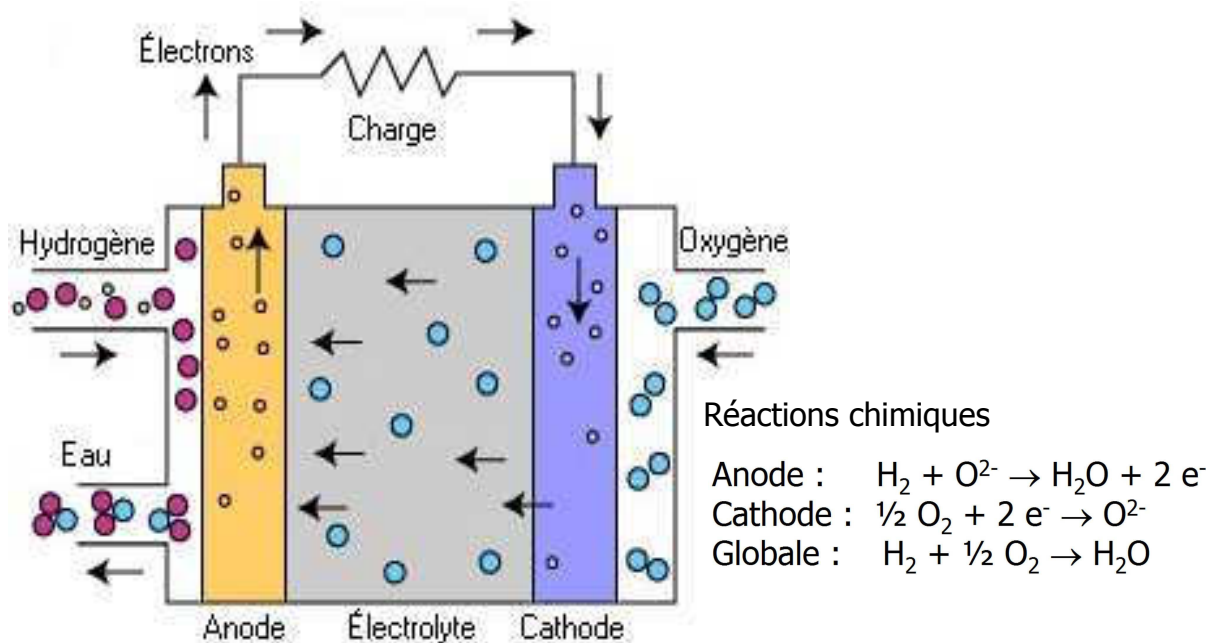


Figure 1.1 : Schéma de principe d'une pile à combustible SOFC (Source : <http://www.fuelcell.org>)

Une pile SOFC utilise de l'hydrogène ou du monoxyde de carbone du côté anodique et de l'oxygène du côté cathodique afin de générer de l'eau et de l'électricité. Les ions O^{2-} sont générés à partir d'oxygène pur au contact avec des électrons provenant de l'anode. Les ions migrent à travers l'électrolyte selon différents mécanismes de diffusion illustrés à la figure 1.2. On distingue 6 différents types de mécanismes de diffusion des ions à travers la matrice électrolytique. Une fois les ions O^{2-} arrivés du côté anodique, ils sont combinés avec les molécules de d'hydrogène, formant ainsi de l'eau qui est extraite sous forme de vapeur. La figure 1.2 présente les mécanismes de transport des ions à travers l'électrolyte (Betoule 1994). Ces mécanismes sont basés sur le principe de diffusion ionique des atomes, permettant ainsi le transfert des ions de la cathode vers l'anode. Le mécanisme de transfert des ions O^{2-} dépend grandement de la géométrie des particules autour des ions qui se déplacent, de la présence des sites interstitiels, ainsi que de la présence de lacunes. Toutefois, il peut survenir un phénomène de conductivité mixte, c'est-à-dire que le transport des ions O^{2-} induit par la conductivité ionique est, en quelque sorte, influencé négativement par la conductivité électrique entraînant les électrons à travers l'électrolyte de l'anode en direction de la cathode (Ishihara, Fukui et coll. 2002; Matsuzaki et Yasuda 2002).

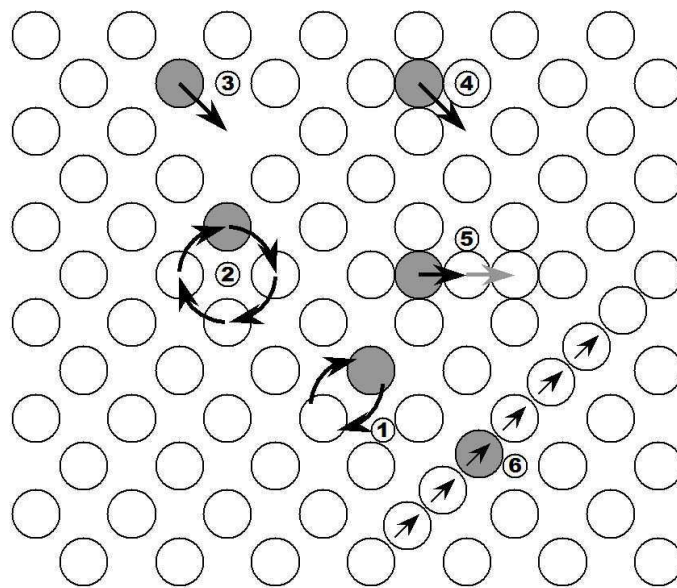


Schéma des principaux mécanismes de diffusion : 1) échange simple; 2) échange cyclique; 3) lacunaire; 4) interstitiel direct; 5) interstitiel indirect; 6) « crowdion »

Figure 1.2 : Schéma des principaux mécanismes de diffusion à travers l'électrolyte (Source : Betoule, 1994)

Une cathode est souvent un composite d'un matériau électrolytique et d'un matériau cathodique. Une couche limite triphasée (TPB) connectée est une section de la couche limite qui est branchée électriquement à chacune des deux surfaces opposées, comme représenté à la figure 1.3. Dans cette figure, la surface supérieure est considérée comme étant la surface électrolytique, la surface inférieure est considérée comme étant la surface cathodique. Les cercles représentent la position des éléments de la structure composite de l'électrode. Le code de couleur permet de distinguer l'électrolyte et la cathode. Le site inactif n'est pas en contact électrique avec les deux surfaces simultanément. C'est pour distinguer ces deux situations que l'algorithme de Hoshen-Kopelman a été adapté en trois dimensions (3D) et utilisé pour les besoins de ce travail.

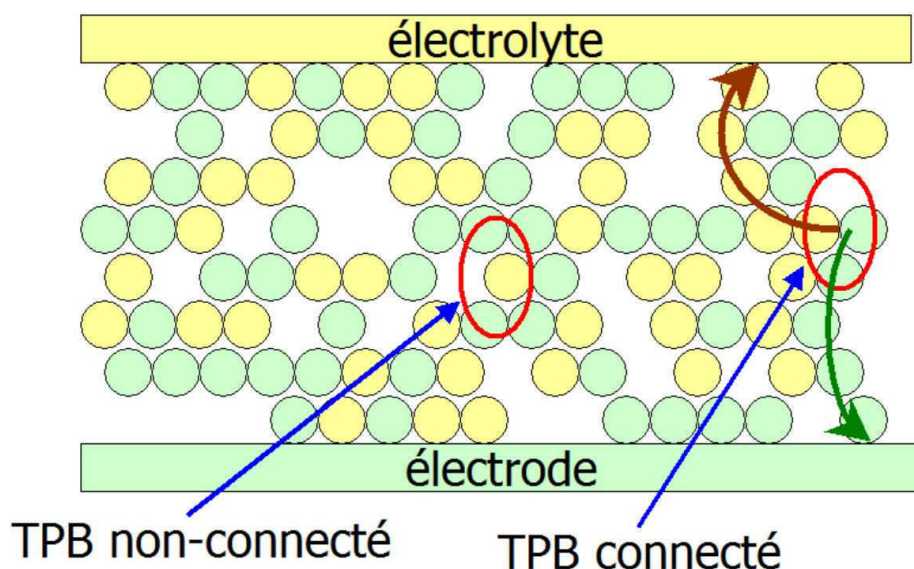


Figure 1.3 : Distinction entre les sites TPB connectés et les sites TPB non-connectés

1.2 Objectifs du projet

L'objectif général de ce projet est de créer une méthode d'analyse permettant de mesurer la longueur de la TPB dans une cathode d'une pile à combustible de type SOFC. Dans le but de réaliser cet objectif, il est nécessaire de développer un protocole visant à obtenir une série d'images par tomographie pour appliquer un algorithme de calcul pour l'estimation de la longueur de la TPB. Les objectifs spécifiques se décrivent ainsi :

- Développer, appliquer et valider un protocole de capture des images obtenues par tomographie.
- Fabriquer des cathodes témoins
- Développer une méthode de création des matrices de calculs incluant le traitement des images.
- Mettre au point un algorithme de calcul Matlab et tester son fonctionnement. (Cet algorithme permet d'utiliser une matrice en 3D qui a été générée à partir de la technique tomographique et de faire le calcul de la longueur de la TPB et sa connectivité.)
- Produire une analyse de directionnalité et de sensibilité afin de s'assurer que les estimations obtenues avec l'algorithme donnent des résultats comparables à ceux obtenus avec un logiciel commercial qui génère des structures en 3D.

Ces objectifs permettront de créer un outil de mesure de la TPB géométrique visant à prédire la résistance au transfert de charge et conséquemment la densité de courant maximale que la pile pourrait générer. Cet outil de mesure permet de vérifier l'influence de la morphologie, la composition en matériaux cathodique et électrolytique et la porosité sur la longueur connectée de la TPB géométrique d'une interface cathode/électrolyte.

1.3 Hypothèse de recherche

L'hypothèse de recherche a été déterminée en fonction de la disponibilité des ressources et des matériaux permis pour l'appareil tomographique. Cette hypothèse de recherche est formulée comme suit :

« On peut mesurer la longueur de la TPB en utilisant la technique de tomographie par faisceau d'ions focalisés (FIB) et le traitement mathématique des images par microscopie électronique à balayage (MEB) avec un algorithme de calcul »

La technique de tomographie a été réalisée sur une cathode formée de gadolinium dopée au cérium (GDC) et de lanthane strontium manganate (LSM). Cette combinaison donne une cathode qui est surtout conducteur électronique et conduit peu les ions (Huang, Schroeder et

coll. 1999), ce qui permet de conclure que le site d'échange principal de la réaction est situé au niveau de la TPB.

La mesure de la longueur géométriquement connectée de la TPB par l'approche tomographique FIB couplée à l'analyse d'images à une résolution supérieure à ce qui connu actuellement dans la littérature constitue une nouveauté de ce doctorat. Une autre nouveauté est que l'outil de mesure qui a été développé dans le cadre de ce doctorat permet de faire la mesure de la longueur de la TPB qui est connectée électriquement avec les surfaces respectives grâce à l'algorithme de Hoshen-Kopelman qui a été adapté en 3D.

CHAPITRE 2

Recherche Bibliographique

2.1 Technique de coupes successives

On recense plusieurs articles concernant des modèles visant à mesurer la longueur de la TPB et la connectivité des TPB. Les modèles utilisés dans la littérature sont constitués majoritairement de particules sphériques de chaque phase (électrolytique et électronique) dans une matrice poreuse.

Plusieurs auteurs ont publié des techniques de gravure avec le FIB dans l'analyse des microstructures. Un groupe (Holzer, Indutnyi et coll. 2004) a présenté les résultats de l'étude 3D de la structure à base de BaTiO_3 par nanotomographie FIB. Les auteurs ont adopté une procédure de coupes successives pour évaluer la structure 3D d'un matériau monophasique imprégné de résine époxy. La résolution obtenue par cette technique était de $714 \text{ nm}^3/\text{voxel}$, ce qui correspond à une résolution d'environ 3 fois plus faible que celle obtenue dans ce doctorat. La porosité a été calculée suivant cette technique FIB et elle a été comparée avec la technique de porosimétrie au mercure (MIP). Les auteurs ont rapporté des différences significatives entre ces deux techniques pour l'estimation de la porosité. Les auteurs ont attribué ces différences au fait que certains pores fermés n'ont pas été pénétrés par du mercure (inkbottle effect). La tomographie FIB a permis d'obtenir des informations quantitatives sur les pores fermés ayant des tailles inférieures à 100 nm, ce qui ne pourrait être obtenu avec d'autres techniques.

Le même groupe de recherche (Holzer, Muench et coll. 2006) a publié un autre travail concernant la tomographie 3D des systèmes de particules. Leur travail a démontré qu'il était possible d'identifier les grains individuels, même dans des structures complexes. Cinq échantillons de ciment ayant des tailles granulométriques différentes ont été imprégnés dans un métal à base de bismuth afin d'exécuter le test tomographique. Les échantillons ont subi une pulvérisation au platine à l'aide d'un plasma afin d'éviter le plus possible le mouvement translationnel (drift) des images durant la procédure de découpage par tomographie. La résolution obtenue était de $1560 \text{ nm}^3/\text{voxel}$ à un grossissement de 35 000 X. Ils ont également

conclu que la correction translationnelle des images a été un facteur important pour la précision de la mesure lors de l'exécution de la tomographie à haute résolution.

Au courant de la même année, un autre groupe de recherche (Wilson, Kobsiriphat et coll. 2006) ont élaboré une procédure de coupes successives par FIB afin de faire la reconstruction 3D de structures anodiques de Nickel-YSZ. L'étude a été basée sur l'analyse de coupes successives de 50 nm d'épaisseur pour un volume total d'échantillon de 105 nm^3 . Chaque coupe tomographique a été suivie par la MEB pour faire la reconstruction de la structure. Les informations de longueur de la TPB, de tortuosité et de porosité ont été mesurées à partir d'un logiciel appelé FEMLAB appliqué sur les images capturées par tomographie.

En 2007, un autre groupe de recherche, dirigé par Gostovic (Gostovic, Smith et coll. 2007), a étudié la région électrochimiquement active d'un pérovskite composé de ferrite lanthane strontium cobalt via la technique de tomographie par FIB. Le volume étudié correspond à $1065 \text{ }\mu\text{m}^3$ à partir de la surface libre jusqu'à l'interface de l'électrolyte dense formée de YSZ. Ils ont mesuré diverses propriétés des microstructures, telles que la porosité globale, la porosité fermée, la tortuosité interfaciale, la longueur de la TPB et la taille des pores. Les auteurs ont aussi présenté les résultats de spectroscopie d'impédance et les données de microstructure.

Un modèle intéressant, paru en 1998 (Wang, Jiang et coll. 1998) concerne un modèle à deux dimensions (2D) basé sur des sphères intercalées formant une matrice électronique pour différents matériaux possédant plus ou moins de conductivité ionique. Les auteurs examinent particulièrement la dépendance de la connectivité effective, de la résistance au transfert de charge et de l'aire de la surface interne en fonction de l'épaisseur de l'électrode, et ce, pour plusieurs combinaisons de matériaux. Selon les auteurs, les résultats électrochimiques semblent correspondre à ceux recensés dans la littérature. Les conditions de modélisation illustrées dans cet article sont à une température de 900°C pour des sphères de 0,1 microns de diamètre. Un résultat intéressant est que l'épaisseur optimale d'une cathode d'erbium dopé à l'oxyde de bismuth / platine pour minimiser la résistance au transfert de charges est de $90 \text{ }\mu\text{m}$ pour une composition de 37% volumique de platine. Lorsque la composition est de 50% volumique de

platine, l'épaisseur optimale de la cathode passe à 10 μm . L'auteur étudie dans cet article deux autres cas d'électrodes : le YSZ / nickel et le YSZ / LSM.

En juillet 2006, un article a paru dans le journal Nature (Wilson, Kobsiriphat et coll. 2006) portant sur la technique de tomographie combinée avec une mesure de connectivité. Les auteurs ont mesuré avec succès la longueur de la TPB d'une matrice anodique formée de nickel et de YSZ sur des dépôts de 50 nm d'épaisseur pour un volume d'échantillon total de 105 mm³. Dans cet article, les auteurs ont aussi mesuré la longueur de la TPB connectée de 10 microns, ce qui représente un rapport de connectivité des sites évalué à 63%, dans un échantillon de 6,0 X 5,2 X 3,4 microns. L'article traite aussi de la mesure du facteur de tortuosité et aussi de l'anisotropie de la structure, ce qu'ils estiment, respectivement à 2% et à environ 15%. Néanmoins, la taille de l'échantillon limite ce type de mesure. Les auteurs ont démontré le potentiel de la technique tomographique pour la mesure de la longueur de la TPB connectée mais aussi de l'homogénéité de la structure.

Un peu plus tard, en 2006, un autre article a paru dans le Journal of Power Sources (Abbaspour, Nandakumar et coll. 2006) concernant un modèle géométrique 2D d'une électrode basée sur des sphères compactées représentant une cathode poreuse de LSM et de YSZ. Les auteurs ont montré l'effet de la composition de la cathode sur la surface de contact entre les phases, sur la longueur de la TPB et sur la résistance à la polarisation de l'électrode. Selon les auteurs, le modèle géométrique proposé semble concorder avec les résultats expérimentaux obtenus dans la littérature. Les auteurs ont aussi démontré que la concentration de l'oxygène dans les zones poreuses de l'électrode est non seulement dépendante de l'épaisseur de l'électrode mais également de sa largeur. L'article conclut en notant qu'à 50% volumique de LSM dans l'électrode, la résistance à la polarisation atteint un maximum. Il est à noter que l'article mentionne spécifiquement la difficulté à tenir compte de la troisième dimension lors l'élaboration de modèles numériques.

En 2010, (Kenney et Karan 2010), un modèle géométrique a été développé concernant une structure 3D composée de sphères enchevêtrées d'une cathode poreuse à base de LSM et de YSZ. Le modèle est utilisé pour déterminer des paramètres électrochimiques importants,

notamment la longueur de la TPB, sa connectivité et l'aire de la surface interne des pores. Ce travail illustre que la longueur de la TPB et sa connectivité est maximale à environ 53% volumique de LSM. Le sujet de cet article traite des modèles de percolation, dont le seuil est défini à un intervalle de porosité situé entre 30% et 40% volumique de LSM. Les auteurs ont aussi déterminé que, pour une structure ayant 25% de porosité avec un diamètre moyen des pores évaluées à 0,19 micron, plus de 99% des pores sont ouverts.

Un article particulièrement intéressant (Lorensen et Cline 1987) propose un algorithme de « marching cube » permettant de faire une visualisation plus précise des surfaces lors des reconstructions 3D issues d'informations tomographiques. Il ne s'agit pas d'un modèle, proprement dit, sur les piles à combustible. Cet article réfère plutôt à une méthode d'adressage des voxels selon un repère bien défini.

D'après ce que nous avons vu dans la littérature, il y a lieu de développer un outil de mesure qui permettra d'innover en matière de résolution tomographique, en matière de la mesure de la longueur connectée de la TPB et de la validation numérique par tomographie virtuelle.

2.2 Photolithographie d'électrodes à motifs

La photolithographie est une technique qui est généralement utilisée pour la fabrication d'électrodes à motifs dans le domaine des piles à combustible. Cette technique est autant utilisée pour la création de cathodes à motifs que pour les anodes à motifs. Elle permet une bonne approximation de la longueur de la TPB car elle permet de créer des patrons aux contours bien définis. Si les électrodes sont fabriquées à partir d'un mélange entre un matériau métallique et un matériau ionique, la mesure est biaisée puisque la réaction électrochimique devient un phénomène volumétrique qui n'a plus rien à voir avec la jonction entre les zones électrolytiques et métalliques. On présente ici quelques articles importants qui serviront de référence au présent document.

Mizusaki (Mizusaki, Tagawa et coll. 1994) a examiné une anode de nickel déposée par photolithographie selon un patron linéaire variant de 5 microns à 100 microns de largeur sur un substrat de YSZ. Les auteurs ont ensuite analysé les anodes par spectroscopie d'impédance en

utilisant de l'hydrogène contenant 4,3% molaire de vapeur d'eau. À des températures au-dessus de 850°C, les phénomènes de frittage du nickel changent substantiellement les résultats d'impédance. Les auteurs ont montré que la résistance au transfert de charge est proportionnelle à la longueur de la TPB. Il faut noter que les résultats préliminaires démontrent que l'impédance obtenue expérimentalement est comparable à celle rapportée dans la littérature.

Bieberle (Bieberle et Gauckler 2000) mesure la longueur de la TPB par l'intermédiaire de la technique de la photolithographie d'anode de nickel sur des électrolytes formés de YSZ ayant une épaisseur 0,5 mm. Les auteurs examinent plus particulièrement la géométrie carrée ou la géométrie formée de lignes parallèles alternant l'anode et l'électrolyte, tel qu'illustré à la figure 2.1. Les anodes ont une surface de 1 cm² et un périmètre qui est mesuré à l'aide de la MEB. Pour chaque cas d'anode étudié, un spectre d'impédance est produit dans le but de trouver une corrélation entre la résistance interne, la densité de courant et la longueur de la TPB. Selon les auteurs, la résistance anodique est inversement proportionnelle à la longueur de la TPB. Les échantillons étudiés dans le cadre de cet article avait une longueur de la TPB variant de 0,04 m/cm² à 4 m/cm².

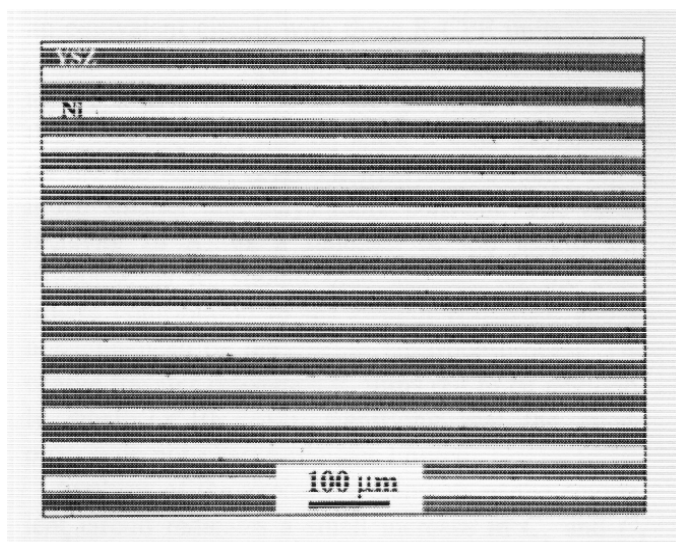


Figure 2.1 : Géométrie anodique parallèle (tirée de l'article de Bieberle, 2000)

Au cours de l'année 2000, le groupe de de Boer (de Boer, Gonzalez et coll. 2000) ont rédigé un article concernant les performances électrochimiques d'une anode constituée de nickel évaporé à la surface d'un électrolyte de YSZ. Les dimensions de l'électrolyte étudié étaient de 16 mm de diamètre et d'une épaisseur de 4 mm avec une densité de $5,89 \text{ g/cm}^3$. Les auteurs comparent la surface occupée par l'anode de nickel avant et après l'analyse de l'impédance. Cette analyse s'échelonnait sur une durée de 50 heures à une température de 850°C faisant varier la fréquence de 1 MHz à 0,01 Hz. Les résultats révèlent que la conductivité change de façon significative lors de la première période de 30 heures et tend à devenir constante par la suite. Les auteurs ont observé que cette tendance fait en sorte qu'il existe une linéarité entre la conductivité et la longueur de la TPB après le test. L'utilisation de seulement 8% de YSZ dans l'anode de nickel tend à augmenter significativement la conductivité de la pile faisant passer la longueur de la TPB de 45 m/cm^3 à 61 m/cm^3 . De plus, on observe que le frittage de la surface de nickel est grandement responsable de la modification de la longueur de la TPB durant le test d'impédance.

Par la suite, un autre groupe a publié deux articles (Radhakrishnan, Virkar et coll. 2005; Radhakrishnan, Virkar et coll. 2005) qui examinent la résistance aux transferts de charge en fonction de la longueur de la TPB à l'aide des cathodes à motifs. Les articles, publiés à quelques mois d'écart, traitent respectivement du LSM et du platine comme matériau de cathode. Les électrodes ont été analysées via une spectroscopie d'impédance afin de déterminer la résistance au transfert de charge en fonction de la longueur de la TPB. Durant le test, les tests d'impédance ont été réalisés avec des températures variant de 650°C à 800°C et des pressions d'oxygène variant de 0,001 atm à 1 atm. Les auteurs ont démontré que la résistance spécifique au transfert de charge (en fonction de la longueur de la TPB) est inversement proportionnelle à la longueur de la TPB. Aussi, les auteurs ont fait l'observation que le résultat de résistance spécifique au transfert de charge est environ 20 fois plus faible pour le platine sur un substrat d'YSZ que pour du LSM sur le même substrat.

2.3 Technique Glycine-Nitrate

La technique glycine-nitrate (GNP) est utilisée couramment pour la fabrication de poudres nanométriques dans plusieurs domaines. Certains auteurs ont utilisé cette technique pour fabriquer des poudres utilisées dans les piles à combustibles.

Le groupe de Chavan (Chavan et Tyagi 2004) ont étudié la GNP pour la production de poudres nanométriques SrCeO_3 et Sr_2CeO_4 . Cette étude a démontré l'importance du rapport oxydant/carburant (O/C) sur la microstructure des poudres produites. Trois types de compositions ont été étudiés, soit : limitant en carburant (O/C = 1 / 1,0), carburant stœchiométrique (O/C = 1 / 1,6) et carburant en excès (O/C = 1 / 3,0). Les résultats ont démontré que la grandeur des particules augmente avec le rapport O/C. Les poudres, une fois formées, s'agglomèrent de façon à former des particules ayant une grandeur située entre 80 nm et 100 nm pour le SrCeO_3 et entre 200 nm et 250 nm pour le Sr_2CeO_4 . L'article démontre aussi l'importance du contrôle du rapport oxydant/carburant sur les poudres obtenues en raison de la stabilité thermodynamique des réactifs.

Les chercheurs du groupe de Peng (Peng, Xia et coll. 2002) ont fabriqué des poudres électrolytiques de samarium dopé au cérium (SDC) par la GNP. Une fois la poudre synthétisée, ils ont fabriqué des pastilles de SDC afin de pouvoir les caractériser. Les auteurs ont étudié différents rapports O/C situés entre 0,7 et 3,4. Ils ont déterminé l'influence de ces poudres sur la morphologie et les propriétés électriques des pastilles. L'article présente des résultats intéressants quant à la conductivité, ils observent que la conductance atteint un optimum lorsque le rapport O/C = 1,7, soit un léger excès de carburant. Aussi, les auteurs ont également démontré que le rapport O/C a une très grande incidence sur la morphologie des poudres tel qu'observée par microscopie à transmission et tel qu'illustré à la figure 2.2.

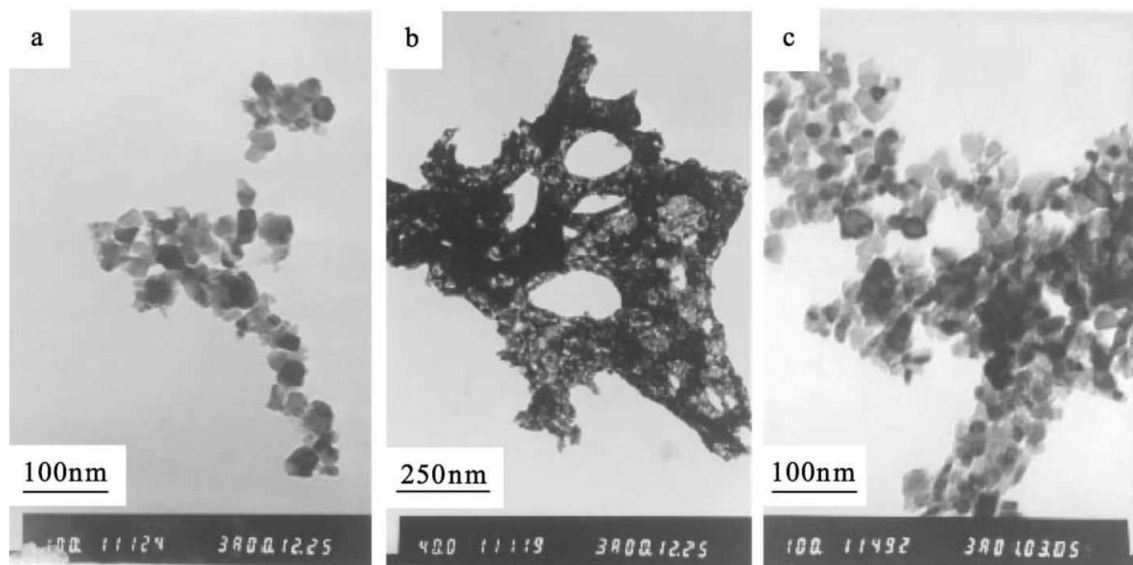


Figure 2.2 : Microscopie à transmission de poudres obtenues par GNP (Source : Peng, 2002)

Mokkelbost (Mokkelbost, Kaus et coll. 2004) a examiné des poudres à base d'oxyde de cérium (soit CeO_2 , SDC et GDC) synthétisé via la GNP. Les poudres d'oxyde de cérium pures (CeO_2) ont été synthétisées via trois différents rapports de glycine/nitrate soit 0,30 (limitant en carburant), 0,55 (carburant stœchiométrique) et 1,0 (excès en carburant). L'article présente les résultats d'analyse thermogravimétrique, de diffraction aux rayons-X, de spectroscopie à infrarouge, d'analyse de surface spécifique et de microscopie à transmission. Chaque échantillon a été calciné à 550°C pendant 12 heures sous présence d'oxygène. L'auteur conclut que lorsque le ratio glycine/nitrate est stœchiométrique, on observe que la taille moyenne des cristaux est d'environ 10 nm et que les poudres qui sont synthétisées sont très réactives et de grande qualité.

2.4 Autres articles importants

L'article qui est très important pour le calcul de la connectivité qui a inspiré ce travail est celui de Hoshen et de Kopelman (Hoshen et Kopelman 1976). Cet article montre une technique d'étiquetage pour calculer la concentration critique, la probabilité de percolation et la taille optimale des particules. L'algorithme qui est présenté fait le balayage des éléments de la matrice de percolation et identifie une adresse à chaque « cluster » (élément occupé) en contact appelé grappes. Chaque grappe d'adresse permet de lier les différentes régions de la matrice et

donc de calculer les caractéristiques de la structure. L'article applique la technique sur une matrice carrée 2D possédant des éléments distribués selon une disposition aléatoire. Ensuite, les auteurs appliquent la technique sur une matrice triangulaire ayant 9 millions d'éléments. Selon les auteurs, les résultats se comparent favorablement aux résultats attendus d'une structure triangulaire infinie.

Metcalf (Metcalf, Kesler et coll. 2009) présentent un modèle numérique basé sur une géométrie à base de sphères calculant la longueur de la TPB et sa connectivité. Les intrants du modèle sont le diamètre des particules, la composition de l'électrode et la porosité totale. C'est par l'entremise de cet article que nous avons travaillé conjointement avec l'équipe de Dr. Kesler de l'Université de Toronto afin de régler le problème des voxels connectés comptés en double. En effet, nos résultats étant très différents de ceux provenant du logiciel Unigraphics, nous avons résolu le problème concernant la somme des voxels. Le filtre a été construit et intégré dans l'algorithme qui est actuellement utilisé pour le calcul de la longueur de la TPB des structures réelles. Dans cet article, des structures ont été générées numériquement afin de pouvoir y mesurer la longueur de la TPB connectée. Notons que la vérification des résultats et la comparaison des résultats de notre simulateur Matlab avec les résultats de logiciel Unigraphics selon les structures fournies par le groupe de Dr. Kesler sont fournies dans ce travail dans la section 4.5.

Nous avons écrit un article dans le Canadian Journal of Chemical Engineering (Rivard, Gitzhofer et coll. 2012) qui présente les grandes lignes du projet d'analyse tomographique. On donne dans cet article la majorité des informations présentes dans le présent document. Cet article présente également une analyse de sensibilité du résultat de la longueur de la TPB en fonction du retranchement des informations selon les trois directions (x,y,z). Cet article est présenté à l'annexe A.

CHAPITRE 3

Description des techniques expérimentales

3.1 Synthèse par la méthode glycine-nitrate

La GNP constitue la synthèse de poudres nanométriques par l'utilisation d'une réaction de combustion entre le glycine (qui constitue le carburant) et le nitrate (qui constitue le comburant) (Chavan et Tyagi 2004). Pour les besoins de la sensibilité de la méthode de calcul, nous avons fabriqué une machine nous permettant de fabriquer de plus grandes quantités de poudres nanométriques. Nous avons créé une cage de synthèse ayant un volume de 1,2 m³ faite d'aluminium muni de deux « canons », tel que représentés sur les figures 3.1 et 3.2. Les « canons » sont, en réalité, deux béciers d'acier comportant un volume maximal d'environ 500 mL. C'est dans ces béciers que le milieu réactionnel est ajouté et qu'un aimant d'agitation est déposé afin de faire chauffer le tout jusqu'à la détonation.



Figure 3.1 : Vue de face de la machine GNP

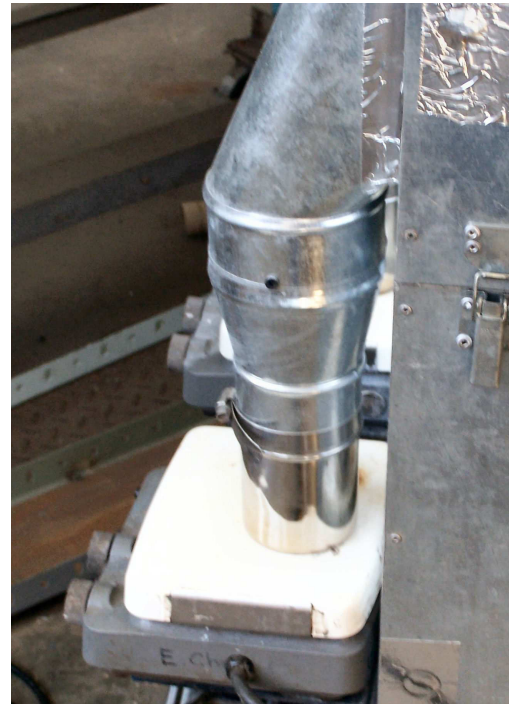


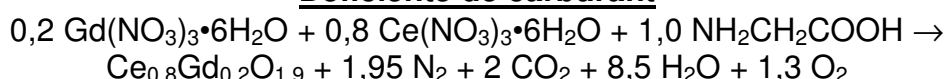
Figure 3.2 : Vue de profil de la machine GNP

La procédure pour la fabrication de nanopoudres par la GNP est relativement simple. Au départ, il faut peser la bonne quantité de nitrates et la glycine. Les quantités dépendent de la taille moyenne des particules voulues, mais surtout de la pression d'explosion. Pour les détails de la recette voir le tableau 3.1. Dans ce projet de recherche, nous avons utilisé un volume de milieu réactionnel de 100 mL pour une production moyenne d'environ 34,5 g de poudres nanométriques à chaque détonation. Nous avons également sélectionné la procédure utilisant le carburant stœchiométrique afin d'éviter d'avoir de la glycine restante dans les poudres récoltées tout en ayant le maximum de conversion des nitrates.

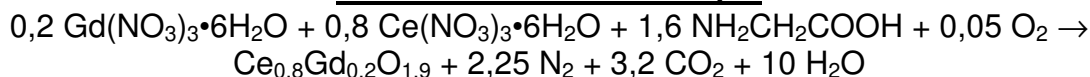
Tableau 3.1 : Paramètres utilisés pour la technique GNP (Source : Chavan, 2004)

Type de procédé	Ratio molaire Oxydant sur Carburant	Masse de nitrate de gadolinium	Masse de nitrate de cérium	Masse de Glycine	Masse de nitrate d'ammonium	Pression d'explosion	Taille moyenne des particules
Déficiance de carburant	1 / 1,0	69,488 g	18,043 g	15,013 g	0,000 g	14470 psi	14 nm
* Carburant Stœchiométrique	1 / 1,6	69,488 g	18,043 g	24,022 g	0,000 g	16244 psi	20 nm
Excès de carburant	1 / 3,0	69,488 g	18,043 g	45,040 g	102,455 g	43591 psi	42 nm

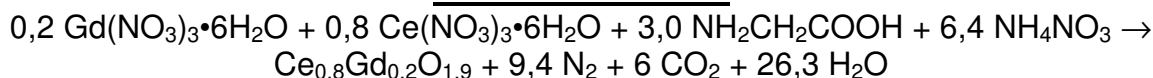
Déficiente de carburant



Carburant Stœchiométrique



Excès de carburant



Les poudres récoltées avaient des structures dendritiques assez complexes, telles qu'illustrées aux figures 3.3 et 3.4. Les poudres post-combustion doivent subir une calcination dans un four à 600°C pendant 2 heures afin d'éliminer toute trace d'humidité assurant ainsi sa conservation

à plus long terme. Une fois les poudres électrolytiques (GDC) et cathodiques (LSM) synthétisées, on doit faire la sérigraphie de la cathode témoin. Cette étape est cruciale, car elle va permettre de produire une cathode réelle pouvant être testée par la technique tomographique. L'utilisation du LSM est due au fait que cette pérovskite minimise les phases magnétiques produites qui pourraient endommager le détecteur de l'appareillage FIB.

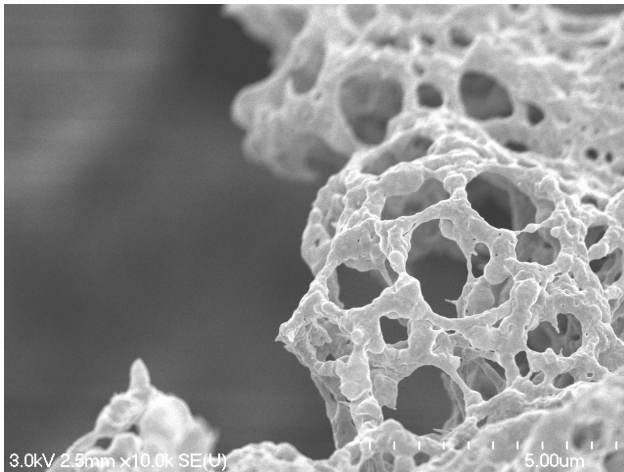


Figure 3.3 : Photo MEB de la poudre GNP (10k X)

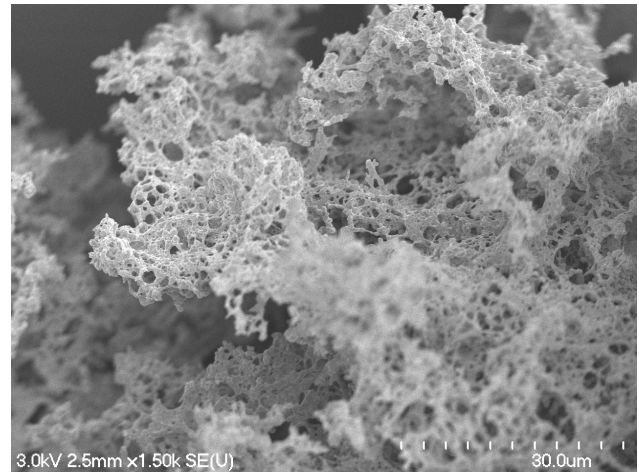


Figure 3.4 : Photo MEB de la poudre GNP (1,5k X)

3.2 Sérigraphie des échantillons de cathode

Le protocole expérimental nécessite la préparation des cathodes poreuses via la méthode de sérigraphie. La sérigraphie utilise le support afin de maintenir l'échantillon en place. On fabrique une pâte cathodique formée des deux principaux constituants (électrolytiques et cathodiques). On applique cette pâte à travers une grille ainsi déposée sur le substrat. Finalement, on referme le couvercle et on passe uniformément la raclette pour faire une surface ayant une épaisseur homogène.

Les échantillons de cathode témoins ont été sérigraphiés sur un substrat de YSZ afin de faciliter l'analyse FIB de la nanostructure. La composition de la cathode est de 50% massique de GDC et de 50% massique de LSM. Les poudres ont été synthétisées selon la méthode GNP présentée dans la section précédente. Afin de générer des pores dans la structure, on a ajouté 10% massique de noir de carbone au mélange et quelques gouttes d'alpha terpinéol afin de créer une pâte visqueuse qui est utilisée pour la sérigraphie. On applique la pâte cathodique sur la grille de sérigraphie à l'aide d'une spatule puis on racle la pâte uniformément. Lorsque la

grille est retirée, un dépôt de pâte d'environ 500 microns d'épaisseur est ainsi créé. La cathode témoin est frittée dans un four à 1200°C pendant 2 heures afin de brûler le carbone laissant derrière les nanopores qui nous permettront de faire la mesure tomographique par FIB. À la figure 3.5, on illustre le résultat de la cathode créée par cette méthode, il faut noter que la porosité est d'environ 35%.

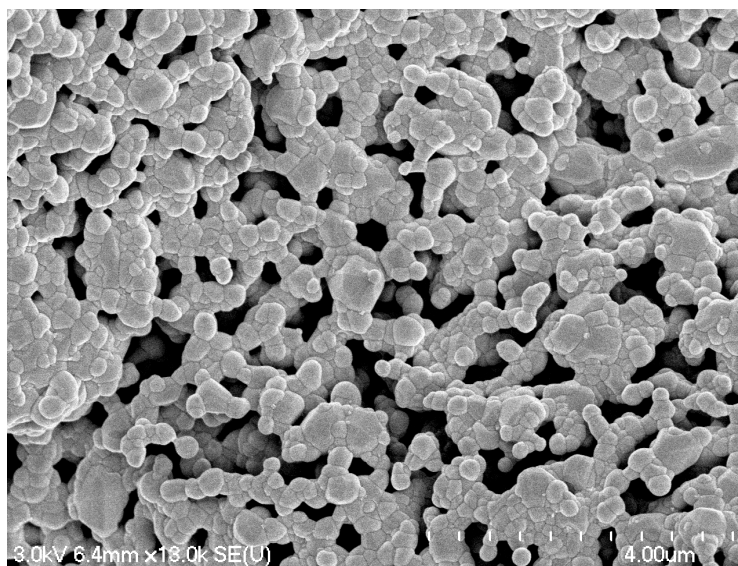


Figure 3.5 : Photo MEB de la cathode déposée par sérigraphie (18k X)

Cette structure constitue le sujet d'analyse de la technique tomographique pour mesurer la longueur de la TPB et la connectivité pour une structure réelle.

3.3 Méthode d'enrobage de l'échantillon

Les échantillons obtenus de la sérigraphie sont imprégnés dans la résine LR White (Electron Microscopy Science, Hatfield PA) selon un protocole modifié. La résine LR White a été choisie en raison de sa faible viscosité et de son aspect facilitant l'enrobage des nanopores. La modification du protocole d'enrobage consiste à favoriser la réaction de branchage des chaînes de polymères. Cette réaction est amorcée par l'ajout de polystyrène et est catalysée par le cérium qui est présent dans l'échantillon. L'enrobage est une étape très importante, car c'est lui qui permet de différencier les phases lors de l'étape de la tomographie. Selon le protocole, l'échantillon à enrober doit être immergé dans 10 mL de résine LR White dans laquelle on ajoute 20 mg de petits morceaux de polystyrène. L'initiateur de réaction est ensuite ajouté dans

le milieu réactionnel et l'échantillon de cathode est déposé dans un dessiccateur à une température de 60°C sous vide. On laisse finalement la résine polymériser au moins 48 heures afin d'atteindre la conversion maximale. À la fin de la polymérisation, l'échantillon est enfermé dans une capsule de polymère ayant une couleur blanchâtre et opaque. La figure 3.6 illustre l'importance du taux de cérium en fonction du milieu réactionnel. De gauche à droite, on illustre 0,1 g de cérium, 0,2 g de cérium et 0,6 g de cérium pour un milieu réactionnel qui contient un échantillon standard contenant des traces de polystyrène. On peut observer que plus le taux de cérium augmente, plus l'opacité du polymère est importante.

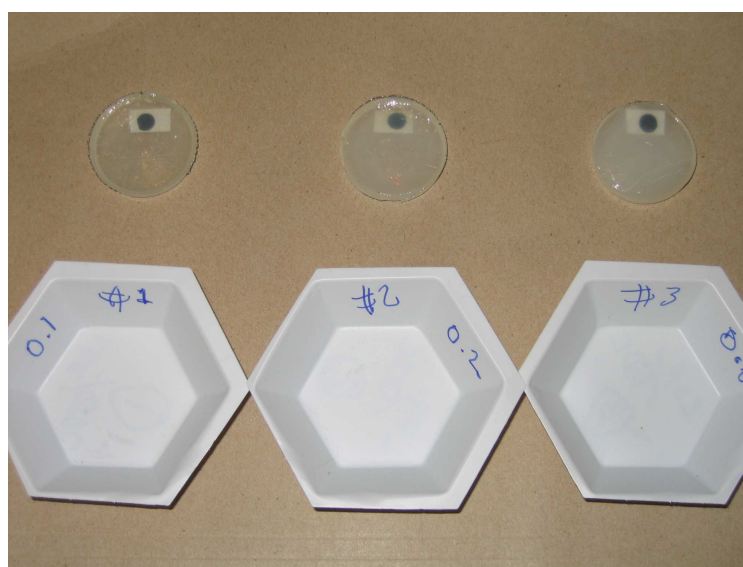


Figure 3.6 : Opacité de l'échantillon en fonction de la concentration de cérium ajouté

Cette analyse démontre l'effet catalytique du cérium dans le milieu réactionnel lors de la polymérisation. Une fois le milieu réactionnel polymérisé et, avant de procéder à la tomographie, la capsule et l'échantillon doivent être cassés en deux et le polymère doit être mécaniquement complètement retiré. On doit polir légèrement la surface afin d'éliminer toute résine résiduelle ce qui aurait pour effet de brouiller les images captées lors de la tomographie. Par la suite, on expose les échantillons à un plasma d'or-palladium pendant 30 secondes pour métalliser l'échantillon afin d'assurer sa conduction lors de l'étape de la tomographie.

3.4 Analyse Tomographique

Le système par FIB peut être employé pour déposer des films métalliques sur un substrat. Pour faire la déposition, on utilise un gaz précurseur organométallique qui est inséré dans la chambre de déposition, habituellement par l'intermédiaire d'une aiguille fine insérée à moins de 100 microns de la position du faisceau d'ions. Le gaz précurseur permet d'entraîner le métal qui sert à la déposition tout en repoussant les produits volatils de la surface du substrat. La capacité de déposer dans des secteurs bien précis du substrat rend les systèmes de déposition par FIB très intéressante pour la nanotechnologie, la nanotechnologie et également pour la préparation des matériaux pour leur caractérisation. Il faut noter que la déposition peut aller jusqu'à 7 nm de diamètre avec une porosité quasi nulle. Certains auteurs ont déjà utilisé cette technologie pour la déposition de métaux sur des surfaces (Davies et Khamsehpour 1996; Kubis, Shiflet et coll. 2004; Lasagni 2008). Cependant, jusqu'à présent, aucun article n'a été répertorié concernant des applications de cette méthode de déposition aux piles à combustibles.

Dans le présent travail, cette technique par FIB est utilisée pour retirer des couches de matériaux afin de générer les images des couches successives nécessaires au calcul numérique de la longueur de la TPB. En effet, une fois la résine retirée, l'échantillon est pulvérisé par un plasma or-palladium afin de créer une fine couche métallique pour rendre l'échantillon conducteur. La procédure de tomographie est réalisée avec un appareil de marque Dual Zeiss 1540XB Crossbeam workstation représenté sur la figure 3.7 qui utilise une source ionique de gallium et qui est muni d'une sonde de coupe de 7,0 nm de diamètre. L'appareil est muni d'un dispositif de MEB utilisant les électrons rétrodiffusés au mode 3-kV de tension d'accélération. La cathode est fixée au socle et mise à la terre à l'aide d'un papier collant d'aluminium. Au départ, le grossissement est faible afin de pouvoir repérer une zone pouvant subir la tomographie et être exempte d'imperfections majeures. On essaie d'éviter les zones possédant des pores importants ou des crevasses afin de ne pas interférer avec les images utilisables pour la tomographie. La capture des images consiste à alterner la gravure par FIB avec la MEB. Les épaisseurs de gravure ont été fixées à $10,0 \pm 0,1$ nm afin d'assurer la qualité des images et une plus grande résolution de la méthode de calcul, c'est-à-dire, une résolution de $250 \text{ nm}^3 / \text{voxel}$.

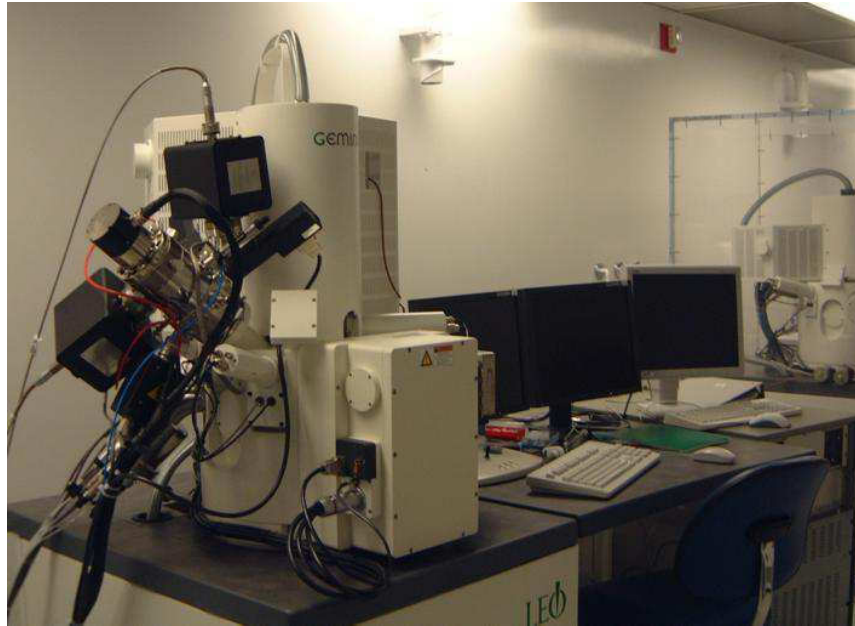


Figure 3.7 : Photo de l'appareillage FIB (Source : CRN2)

Les photos des coupes tomographiques qui sont obtenues par cet appareil sont ensuite acheminées au traitement d'images utilisant le logiciel commercial JASC Paintshop Pro pour l'ajustement des contrastes et la clarification des images, dont le résultat est illustré à la figure 3.8.

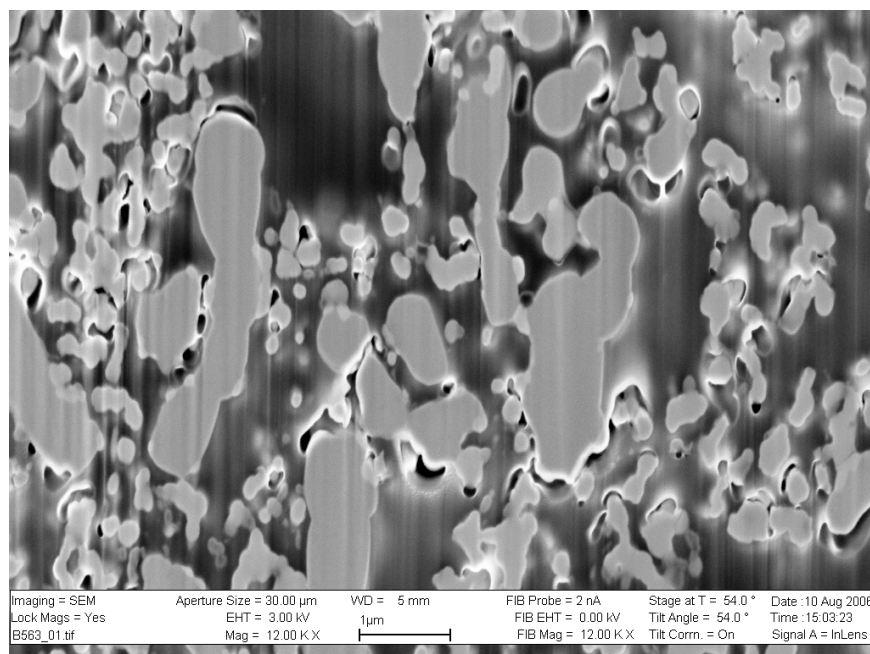


Figure 3.8 : Photo d'une image tomographique d'un échantillon de cathode (12k X)

Un total de deux images sont générées numériquement à partir de chaque coupe tomographique, une pour chaque matériau constituant de la cathode. Ces photos seront transformées en matrices pour le calcul de la longueur de la TPB. Lors de la classification des différents matériaux lors d'une coupe tomographique, il faut s'assurer que l'échantillon soit tomographiable, c'est-à-dire, que les deux matériaux puissent être distingués via une simple troncature des tons de gris, tels que représenté à la figure 3.9. Lors d'une représentation du nombre de voxels en fonction de la valeur du ton de gris, illustrée à la même figure, on peut noter que l'échantillon tomographiable possède deux sommets qui peuvent être séparés par une simple déconvolution. Le résultat du ton de gris frontière entre les deux sommets sera déterminé comme étant la valeur qui permettra de distinguer les deux phases principales lors de la production des coupes tomographiques. Puisque les valeurs de contraste et de luminosité ne changent pas, cette valeur frontière demeurera la même pour chaque image durant la totalité de la tomographie.

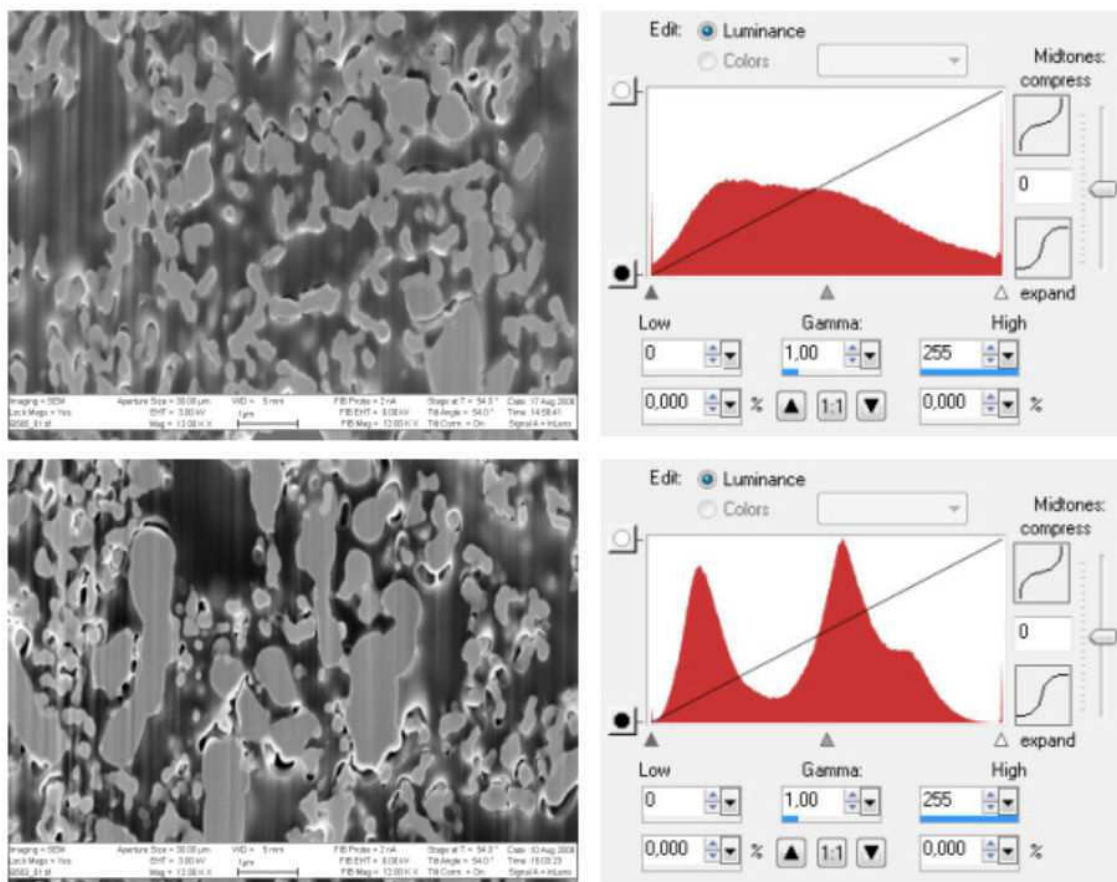


Figure 3.9 : Exemple d'un échantillon d'image tomographiable et non tomographiable

Dans le même ordre d'idée, il faut noter que l'effet translationnel lors la capture des images a comme conséquence une diminution de la taille de l'échantillon. L'effet translationnel est le mouvement horizontal (selon x) ou vertical (selon y) incontrôlé durant la capture d'image. Ce mouvement peut être généré par la concentration des charges sur l'échantillon ou le mouvement vertical dû à l'avancement des coupes. Puisque la capture s'effectue à un angle de 52 degrés, une coupe résulte en un mouvement de l'image vers le haut. La figure 3.10 illustre un schéma du système de coupe et de capture d'images.

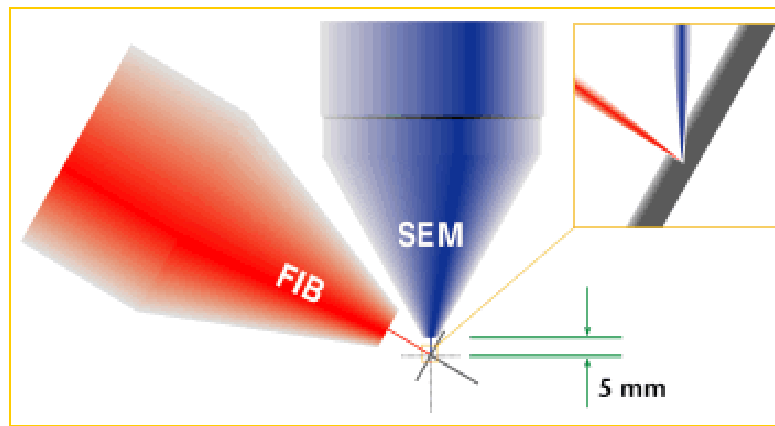


Figure 3.10 : Schéma du système de coupe et de capture d'images

La première étape de la transformation des images est d'éliminer les zones de charges des images (zones blanches). Ensuite, on peut séparer les différents constituants en fonction du ton de gris. En effet, les zones les plus foncées (noires) de cette photo sont représentées par les porosités. Les zones les plus claires (grises pâles) sont représentées par le matériau électrolytique ayant la plus grande masse molaire. Les zones intermédiaires sont représentées par le matériau cathodique. On illustre la démarche pour la séparation des images en matrices à la figure 3.11. Il faut noter que par rapport à l'échantillon d'image initiale obtenu par tomographie, on perd 35% de la surface analysée sans remettre en question la qualité de l'analyse en raison du déplacement translationnel.

3.5.1 Problème de correspondance

Le problème de correspondance est un problème relié à une quantité d'information insuffisante quant aux nombres de tranches d'un artefact donné. Cette carence d'information peut mener à une interprétation erronée d'une structure donnée. La figure 3.12 illustre le problème de correspondance relié à 4 coupes successives.

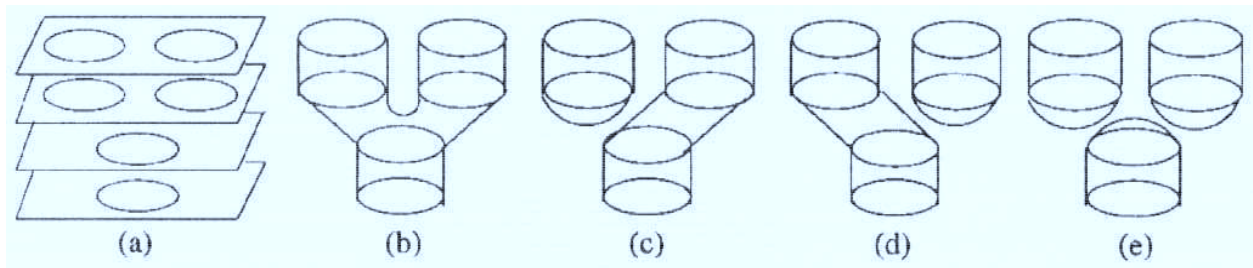


Figure 3.12 : Illustration du problème de correspondance (Source : Bajaj, 1996)

La première figure donne le résultat de la capture d'information à partir de 4 coupes successives. Les 4 autres figures donnent les connexions présumées à partir du résultat de la coupe tomographique. On ne peut donc pas conclure que les coupes obtenues par tomographie proviennent de la même structure. Dans le présent travail, il est important que les coupes successives soient suffisamment rapprochées pour éviter les problèmes de correspondance, car l'étude que nous avons réalisée vérifie la connexion électrique entre deux points d'une structure 3D complexe.

3.5.2 Problème de pivot

Le problème de pivot est défini comme étant de l'information erronée qui est capturée non pas sur la face de la coupe, mais sous la surface de coupe. Ce problème est fréquent si la tension d'accélération du microscope servant à capturer l'image est trop importante. La figure 3.13 illustre une capture d'image avec une tension d'accélération trop importante (30 kV). On peut noter que les structures semblent visibles à l'intérieur de la face coupée. Dans le présent document, il faut utiliser la tension d'accélération la plus faible possible pour pouvoir distinguer les matériaux tout en détectant les électrons à la surface de la structure. Une tension d'accélération de 5 kV pour la capture a été utilisée en choisissant le détecteur des électrons

rétrodiffusés. De cette façon, les informations très près de la surface de coupe sont traitées en priorité et non les structures sous la surface.

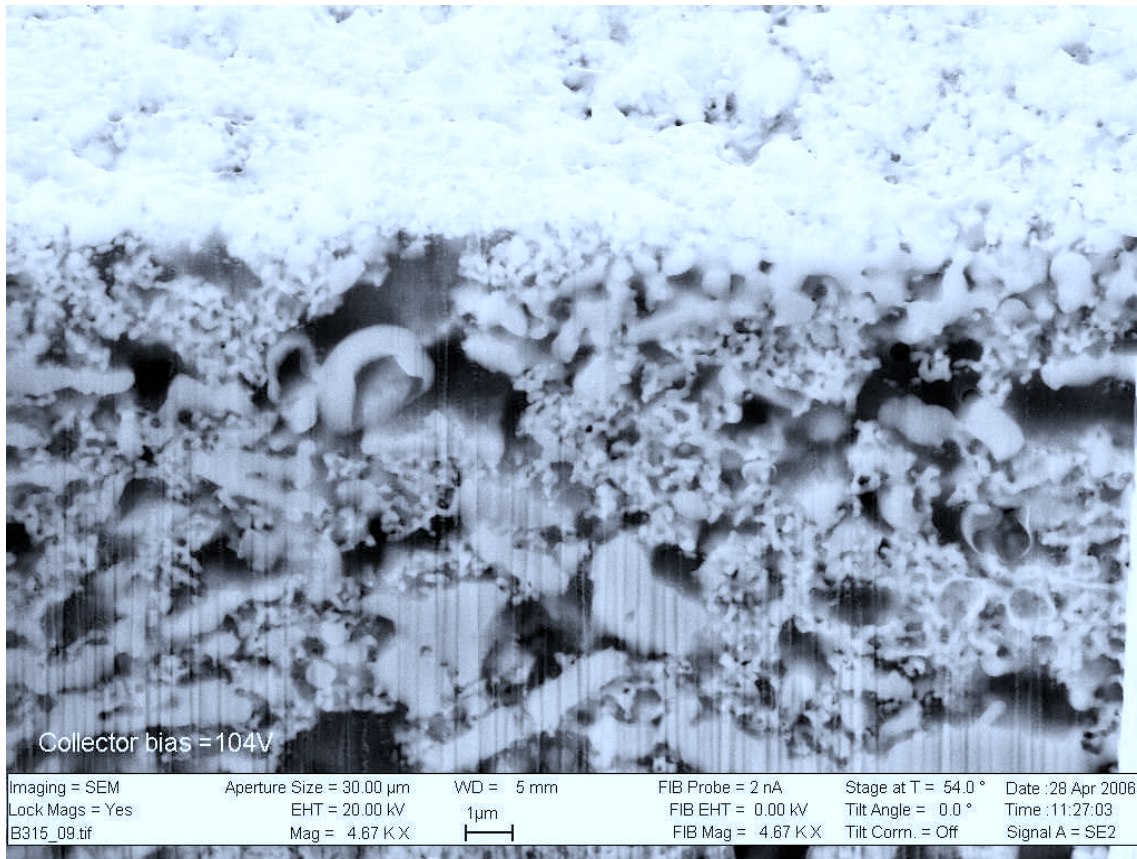


Figure 3.13 : Photo d'une coupe illustrant le problème de pivot

3.5.3 Problème de branchement

Le problème de branchement constitue un problème qui ressemble beaucoup à celui de la correspondance. La figure 3.14 illustre le problème de branchement dans le cas de deux surfaces successives obtenues par tomographie. Dans le cas où C1, C2 et C3 constitueraient la même structure, on ne peut pas être précis sur la position exacte de la jonction L de la structure, et ce, en raison du manque d'information. Les auteurs suggèrent que 4 images par artéfact soient nécessaires pour minimiser les problèmes de correspondance et les problèmes de branchement. Dans ce contexte, on a estimé que, puisque les plus petits agglomérats ont une grandeur d'environ 80 nm, il est opportun d'avoir des tranches de moins de 20 nm.

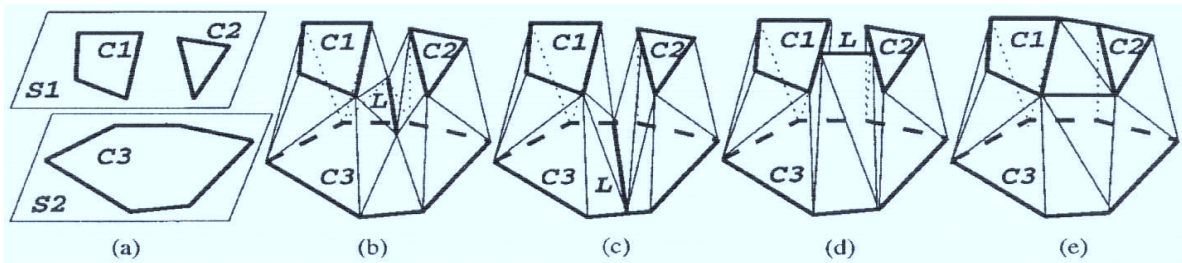


Figure 3.14 : Illustration du problème de branchement (Source : Bajaj, 1996)

3.5.4 Problème d'effet rideau

Le problème d'effet rideau survient lorsque les ions servant la gravure sont réfractés à la surface de l'échantillon. Le problème s'accroît au fur et à mesure que le processus de gravure avance. Lors des expériences préliminaires de tomographie, nous avons observé que lorsque l'espacement entre les tranches est trop faible, les ions de gallium servant à la gravure rebondissent sur la surface. Ce phénomène, appelé l'effet rideau, a pour effet de créer des « vagues » lors de la capture d'image. L'effet est de plus en plus amplifié au fur et à mesure que la tomographie avance. Nous avons également observé que l'effet rideau peut être atténué lorsque l'espacement entre les tranches est augmentée. La figure 3.15 illustre une coupe tomographique à laquelle l'effet rideau est important.

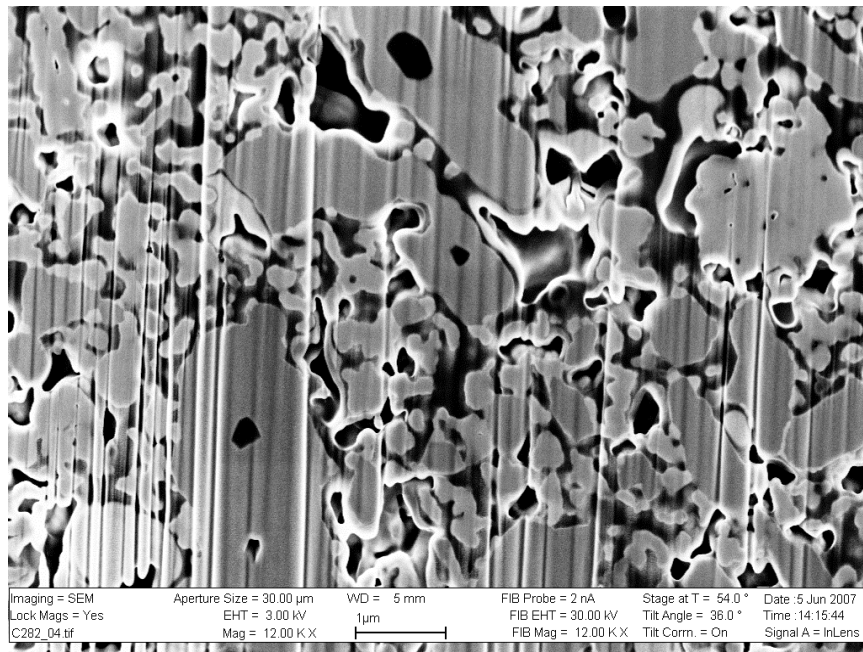


Figure 3.15 : Photo d'une coupe illustrant le problème d'effet rideau

3.6 Photolithographie

La déposition par photolithographie consiste à enduire sur un substrat, un liquide épais qui est soluble à la lumière appelé la résine photosensible. Ce liquide contient du matériel à enduire (Ex : Platine) en solution avec une substance photosoluble. Un masque est utilisé pour couvrir les zones de la surface bloquant ainsi leur exposition aux rayons ultraviolets (UV) qui sont utilisés pour solubiliser la résine. Dans les zones transparentes du masque, la lumière traverse et expose le vernis photosensible, ce qui a pour conséquence de solubiliser la résine ne laissant que le matériel sur la surface. Le vernis photosensible durcit et devient imperméable une fois exposé à la lumière UV. Une fois le processus complété, on procède au nettoyage de la surface avec une solution diluante afin d'enlever le vernis photosensible non exposé. Le résultat d'une telle déposition est une pellicule mince et dense ayant le modèle exact du masque. On donne les étapes du processus photolithographique dans la figure 3.16. La figure 3.17 illustre le modèle du masque utilisé pour créer la courbe de calibration de notre outil de mesure.

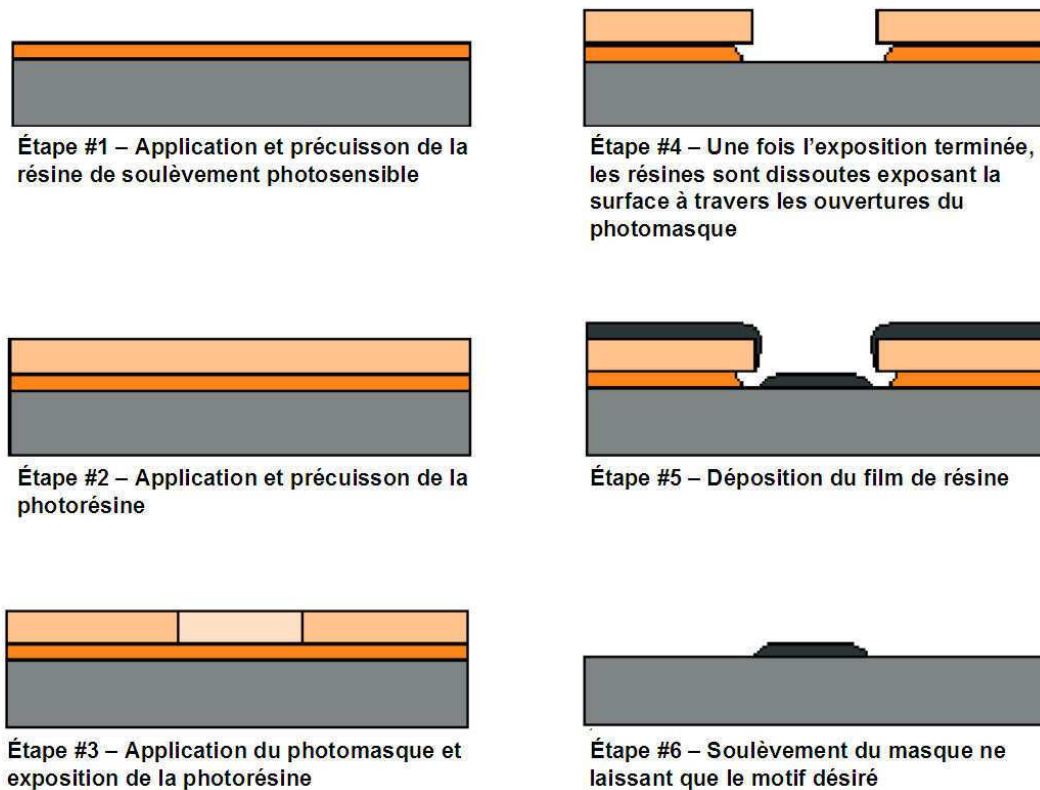


Figure 3.16 : Illustration qui définit les étapes d'une photolithographie

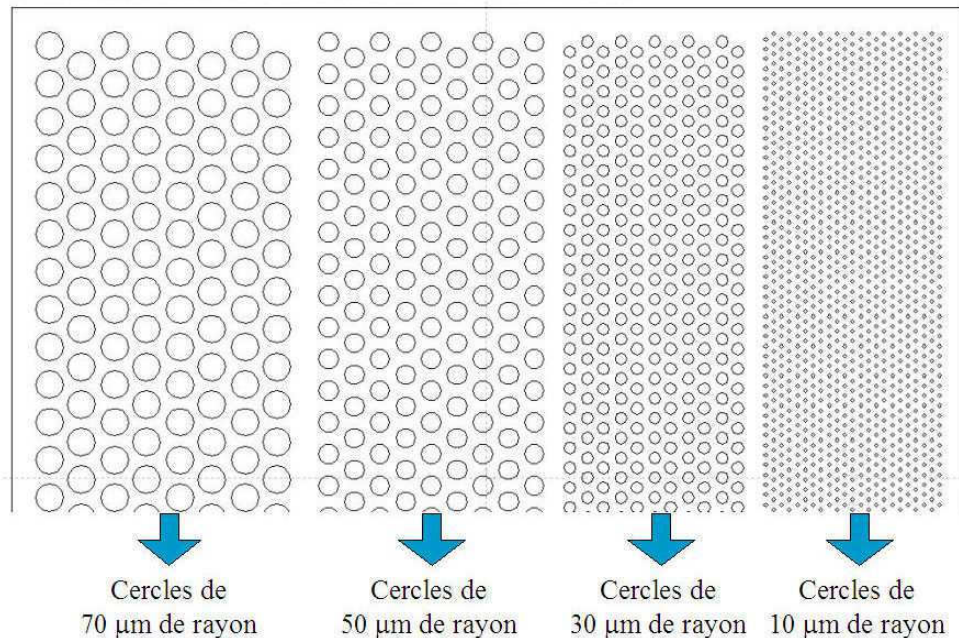


Figure 3.17 : Masque utilisé pour la fabrication de la courbe de calibration

3.7 Pulvérisation cathodique magnétron

La pulvérisation cathodique magnétron est une technique qui consiste à recouvrir la surface d'un substrat par des atomes provenant du bombardement d'une cible (cathode). L'avantage de cette technique est que la déposition peut se faire uniformément sans avoir besoin de faire varier la position du substrat. Ajoutons que grâce à une faible température de dépôt (80°C), il est possible de déposer sur le substrat sans trop désagréger les résines photolithographiques. L'épaisseur minimale de ce type de déposition est de l'ordre de quelques nanomètres avec une porosité quasi nulle et la diffusion du plasma permet un recouvrement uniforme de la périphérie des pièces de géométrie complexe. Les sources de pulvérisation sont habituellement des magnétrons qui utilisent des champs électriques et magnétiques intenses pour emprisonner des électrons près de la surface du magnétron (la cible). Le plasma est maintenu par l'ionisation provoquée par les électrons secondaires émis de la cathode due au bombardement d'ions qui sont accélérés dans le plasma (Srivastava, Quach et coll. 1997). La figure 3.18 donne le schéma de principe de la déposition par pulvérisation cathodique magnétron. La déposition a été effectuée avec une puissance RF sur la cible (LSM) de 115W à une température de substrat de 400°C sous une pression de 5,6 mTorr utilisant un plasma d'argon et d'oxygène.

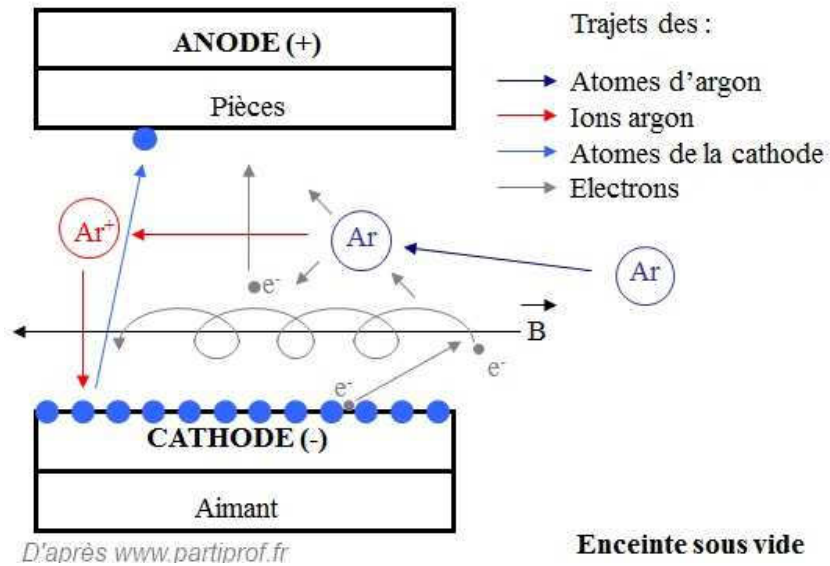


Figure 3.18 : Schéma de principe de la pulvérisation cathodique magnétron (Source : www.partiprof.fr)

3.8 Spectroscopie d'impédance électrochimique

La spectroscopie d'impédance électrochimique constitue un outil diagnostique permettant de caractériser les limites et améliorer la performance d'une pile à combustible. Cet outil de mesure permet de cataloguer les sources de polarisation selon trois catégories : Les pertes cinétiques (transfert de charges), les pertes Ohmiques (transport des ions et des électrons) et les pertes liées au transfert de masse. Le spectre d'impédance sert à déterminer les différentes sources de polarisation par l'intermédiaire d'un circuit électrique équivalent. On illustre à la figure 3.19, les différents modèles de circuits équivalents utilisés dans la spectroscopie d'impédance d'une pile complète (Lvovich 2012).

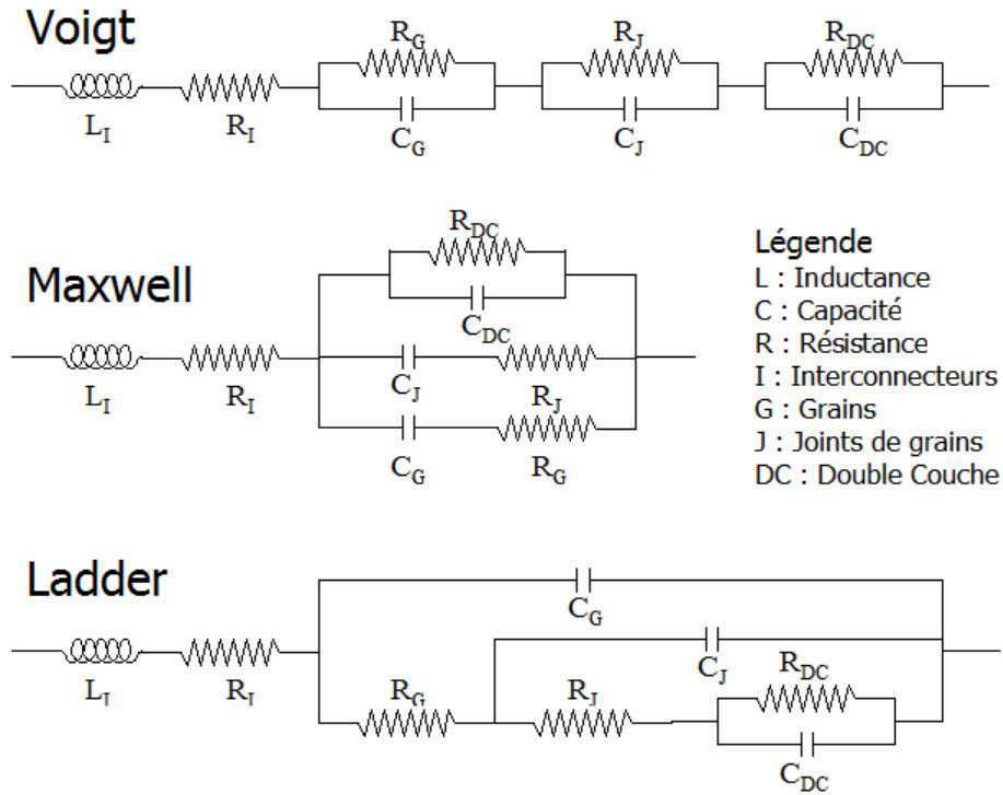


Figure 3.19 : Modèles de circuit équivalent utilisés en spectroscopie d'impédance

Lors d'un test visant à produire un spectre d'impédance, on balaye les fréquences afin de déterminer le comportement de la pile complète. On illustre alors cette courbe de réponse en fréquence sur un diagramme de Nyquist. La figure 3.20 représente un exemple de spectre d'impédance pour une électrode.

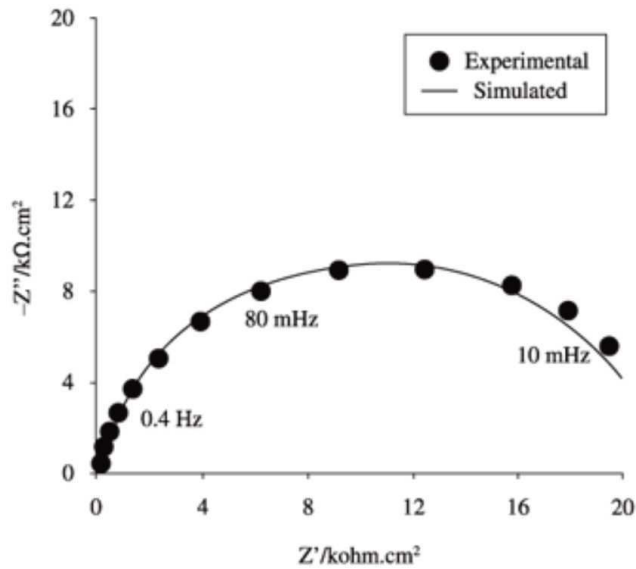
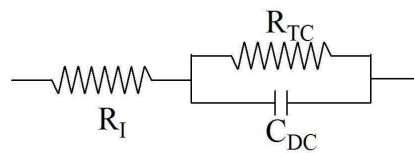


Figure 3.20 : Exemple de spectre d'impédance

Dans le présent travail, l'analyse s'est effectuée sur une cathode seulement. Il est donc opportun de simplifier le modèle de circuit équivalent puisque certaines informations du spectre peuvent s'avérer superflues. Pour l'obtention d'un spectre d'impédance pour une électrode, le modèle le plus couramment utilisé est celui de Voigt simplifié. La figure 3.21 illustre le circuit équivalent de Voigt simplifié.

Voigt simplifié



Légende

- C : Capacité
- R : Résistance
- I : Interconnecteurs
- DC : Double Couche
- TC : Transfert de charge

Figure 3.21 : Modèle de circuit équivalent de Voigt simplifié

Il faut noter que la courbe de calibration a été calculée à partir de la résistance au transfert de charge. La résistance au transfert de charge est représentée dans un spectre d'impédance comme étant la largeur intégrale du spectre, c'est-à-dire le point de croisement entre l'origine et la valeur maximale possible sur l'axe réel (l'axe Z'). Lorsque la courbe ne touche pas l'axe réel, on doit faire le lissage du demi-cercle correspondant afin de déterminer la valeur au point de croisement sur l'axe réel (Virkar, Chen et coll. 2000).

3.9 Analyse par électrode de contact

L'analyse par électrode de contact consiste à utiliser une sonde d'analyse que l'on met en contact avec une électrode à motifs. Étant donné la taille des micro-électrodes, on doit utiliser un microscope optique afin de pointer la sonde sur la bonne électrode (Brichzin, Fleig et coll. 2002). L'analyse se fait par balayage des fréquences afin de générer un spectre d'impédance pour chaque électrode. L'analyse dure environ 30 secondes par électrode et permet de déterminer le résultat de résistance au transfert de charge qui découle de l'analyse du spectre d'impédance de l'électrode. La figure 3.22 illustre le schéma de principe de l'analyseur de microélectrodes à motif.

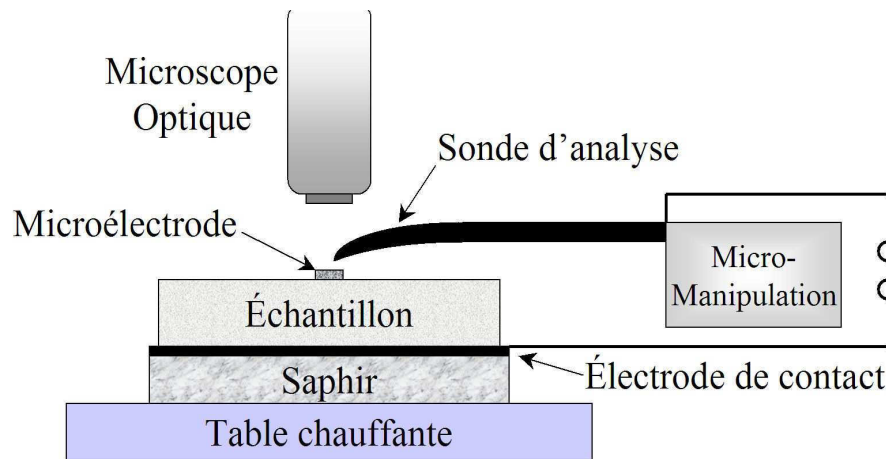


Figure 3.22 : Schéma de principe de l'analyseur de microélectrode à motif

CHAPITRE 4

Analyse numérique et résultats expérimentaux

4.1 Transformation des images en matrices

Le calcul de la longueur de la TPB a été fait à partir de l'empilement d'images à l'aide d'un algorithme de calcul compilé par le logiciel Matlab de Mathworks. Les photos résultant du processus de traitement d'images sont converties en images binaires (en noir et blanc). Le but ultime de cette conversion est de transformer toutes les images en une matrice binaire 3D. Il y a, au total, 2 matrices en 3D, une pour chaque matériau de la cathode. Ce sont ces matrices qui sont utilisées pour l'application de l'algorithme de calcul Matlab. Il est à noter que lors de la conversion des images en matrices, il y a transposition de matrices lors de la lecture des données dans l'algorithme Matlab. En effet, les données (x,y) sont transposées en données (y,x) . Ce changement a été pris en compte lors de l'écriture de l'algorithme de calcul. Mentionnons que, dans le présent travail, il faut définir le terme voxel comme étant un pixel volumétrique ou encore un pixel en 3D.

4.2 Génération des matrices tridimensionnelles

La première étape de la compilation est la lecture des matrices MAT_GDC et MAT_LSM provenant de la conversion d'images en matrices 3D. Les matrices ayant des dimensions importantes sont lues selon un mode binaire, c'est-à-dire une matrice ne comportant que des 0 et des 1. Une fois la compilation réalisée, on crée des matrices d'expansion en ajoutant un voxel supplémentaire au périmètre de toutes les formes de chacune des matrices générant ainsi les matrices d'expansion EXP_GDC et EXP_LSM. Par après, on crée des matrices de contour en faisant la soustraction numérique des matrices expansions avec les matrices de données. En effet, la matrice contour du GDC est obtenue en soustrayant EXP_GDC avec la matrice MAT_GDC. De la même façon, la matrice contour du LSM est obtenue en soustrayant la matrice EXP_LSM avec la matrice MAT_LSM. Toutes ces matrices sont importantes pour les étapes de calcul subséquentes menant au calcul de la longueur de la TPB. On illustre dans les figures 4.1 et 4.2 la méthode d'expansion des matrices électrolytiques et électroniques. On peut observer que l'expansion selon les faces donne des résultats qui peuvent faire en sorte que

certaines informations soient perdues durant le processus. Nous avons adopté une expansion selon les faces et les coins afin d'inclure le maximum d'informations. Nous avons dû construire un filtre afin d'éliminer les voxels comptés en trop.

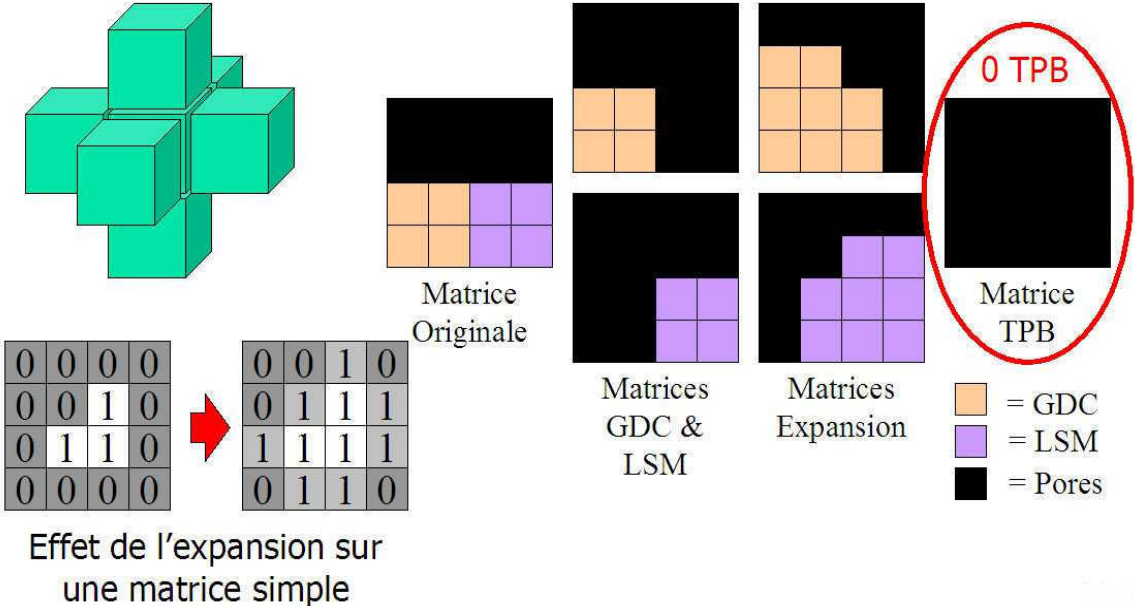


Figure 4.1 : Illustration représentant l'expansion matricielle selon les faces

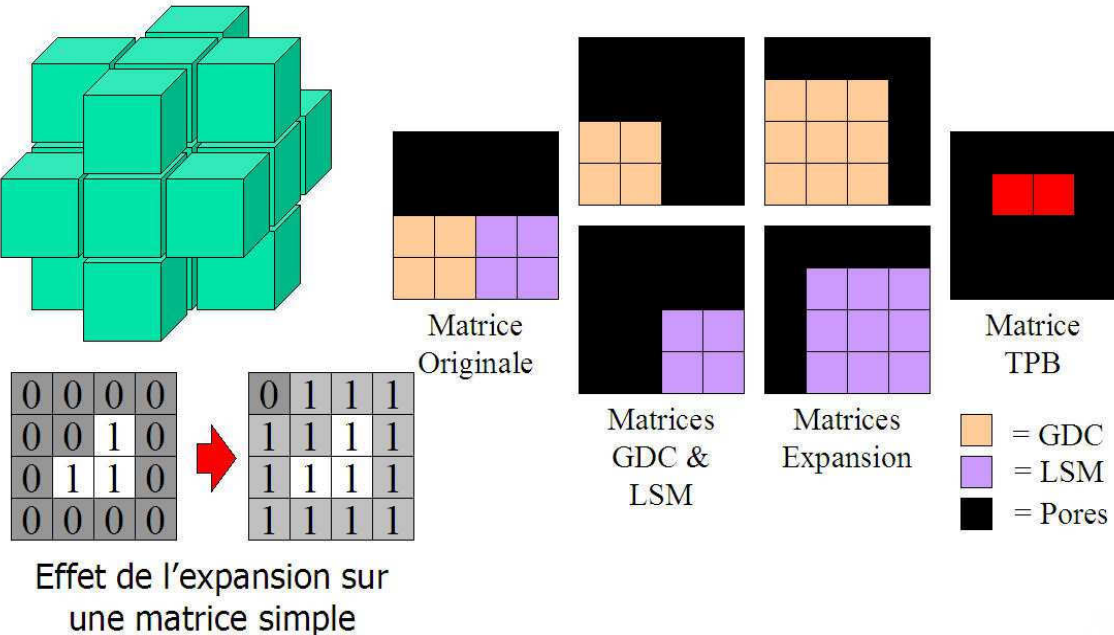


Figure 4.2 : Illustration représentant l'expansion matricielle selon les faces et les arêtes

4.3 Description du filtre numérique

Dans une expérience qui a été réalisée sur une matrice numérique en forme de demi-cercle continu de 200 pixels de rayon dans le plan XY pivoté à 45 degrés selon l'axe des Z qui contenait un total de 630 voxels, nous avons appliqué l'expansion suivant les faces seulement. L'algorithme a trouvé 357 voxels. On peut conclure que l'algorithme d'expansion fait en sorte que de l'information est perdue lors du traitement. Lorsque nous avons appliqué l'algorithme suivant une expansion sur les faces et les arêtes, 948 voxels ont été trouvés. Donc, nous avons prouvé, par ce petit test, qu'il y a des voxels comptés en trop dans l'algorithme d'expansion des faces et des arêtes. De ce fait, nous avons dû construire un filtre numérique pour éliminer certains voxels.

Le filtre numérique permet d'éliminer les voxels comptés en trop dans la matrice. Puisque l'expansion selon les faces et les arêtes a été retenue, il nous fallait trouver une solution quant à l'élimination des voxels excédentaires. Ce filtre détermine la direction principale des voxels dans une région donnée et élimine les doublons dans la direction qui n'est pas la direction principale de la ligne. Ce filtre numérique s'assure que les voxels qui demeurent dans la matrice TPB sont en contact avec, au minimum, une arête afin d'assurer la continuité électrique. Les figures 4.3 et 4.4 montrent une partie de la matrice TPB avant et après le passage du filtre. On peut remarquer qu'au moins 30% des voxels ont été retirés, étant jugés excédentaires. Ces voxels excédentaires proviennent du résultat de l'expansion numérique sur les faces et les arêtes montré à la figure 4.2. Afin de tester son efficacité, nous avons appliqué le filtre sur le cas du demi-cercle illustré au début de cette section et le nombre de voxels est passé de 948 à 690.

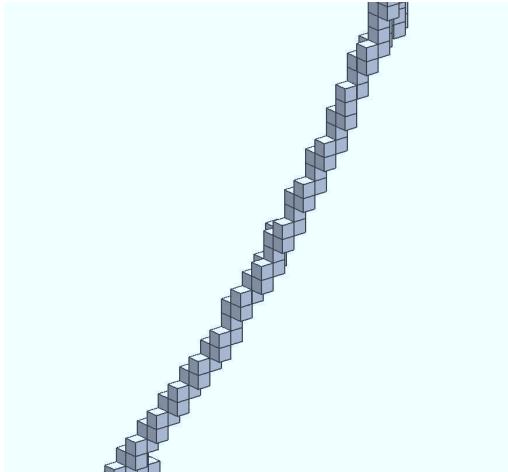


Figure 4.3 : Représentation des voxels avant le filtre

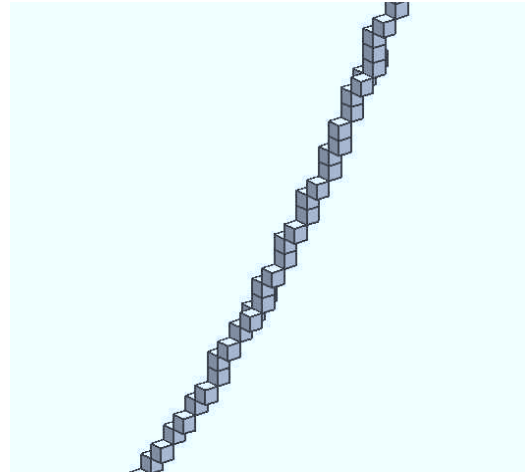


Figure 4.4 : Représentation des voxels après le filtre

4.4 Mesure de la couche limite triphasée

La matrice représentant la TPB appelé MAT_TPБ est le résultat de la multiplication des deux matrices contours de chaque matériau. On calcule ensuite le nombre de valeurs non nulles de la matrice résultante via une sommation de toutes les données afin de déterminer le nombre de voxels. Le nombre total de voxels est simplement multiplié par la longueur caractéristique du voxel calculé par un sous-programme. La longueur obtenue représente une estimation de la longueur de la TPB en fonction de l'échantillon mesuré au départ. Notons que la sensibilité dépend grandement du grossissement utilisé et surtout du succès de l'enrobage lié au choix de la bonne résine. Il faut noter que la mesure obtenue après ce procédé est notée en terme de voxels par voxels cubes. Cette mesure n'a pas de signification si le grossissement et la résolution ne sont pas connus. En conséquence il faut calculer la longueur équivalente à un voxel.

4.5 Calcul de la longueur caractéristique d'un voxel

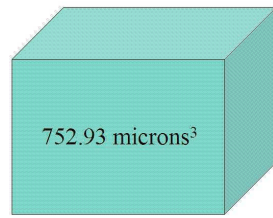
Il faut déterminer la taille réelle d'un voxel afin de pouvoir la convertir en grandeur métrique. Il faut faire la transformation à partir des unités de voxels par voxels cube en nanomètres par nanomètres cube. Pour réaliser cette opération, des tests ont été exécutés avec l'algorithme Matlab en parallèle avec les résultats provenant d'un autre logiciel appelé Unigraphics. Le logiciel Unigraphics a permis d'obtenir des résultats de longueur sur des structures complexes

composées de sphères. Ces résultats ont servi de référence aux calculs réalisés par l'algorithme Matlab.

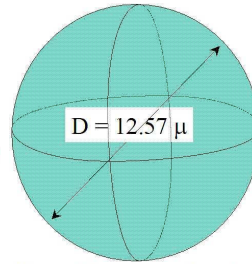
Par la suite, nous avons utilisé la méthode d'attribution d'une longueur équivalente au voxel. La démarche expérimentale utilisée consiste à remplacer les voxels par des entités géométriques afin de calculer la longueur équivalente au voxel. Nous avons étudié 5 différentes façons d'additionner les entités géométriques, chacune donne des résultats différents. Subséquemment, nous avons comparé le résultat obtenu avec le résultat obtenu du logiciel Unigraphics selon la même géométrie. Il est important de noter que le voxel possède une géométrie de type parallélépipédique (non cubique). La mesure de chaque côté dépend de la mesure de l'échantillon tomographique tout en tenant compte de la distorsion.

Le premier cas consiste à calculer le diamètre de la sphère possédant la même surface extérieure que celui du voxel. Une fois le diamètre de la sphère équivalente connue, il ne suffit que de le multiplier avec le nombre total de voxel dans la matrice TPB. Le deuxième cas consiste à utiliser une géométrie de type tuboïde ayant la même surface extérieure que le voxel. Cette géométrie tient compte de la distorsion advenant que le voxel ne serait pas cubique. Il faut aussi noter que la distance entre le centre du tuboïde et le point de contact avec le voxel voisin sont additionnées. Le résultat de la somme est divisé par le nombre de segments. Si un segment est continu et traverse de part en part le tuboïde, on tient compte de la longueur totale. Le troisième cas constitue à déterminer la distance centre-à-centre entre deux voxels adjacents. Lorsque plusieurs voxels sont impliqués, la distance est moyennée entre chacun des voxels voisins. Dans le cas d'une ligne, la distance est simplement doublée afin d'obtenir une longueur caractéristique pour chacun des voxels de la structure. Le quatrième cas ressemble au premier cas, car il consiste à calculer le diamètre de la sphère possédant le même volume que le voxel. Le calcul du volume est basé sur l'approximation que le voxel est sphérique. Le cinquième et dernier cas ressemble au deuxième, car il consiste à utiliser une géométrie de type tuboïde ayant le même volume que le voxel. Les mêmes calculs sont utilisés quant aux voxel voisins. Les cas cités en exemple qui sont illustrés à la figure 4.5 correspondent à un voxel de taille 9,2 microns de hauteur x 9,3 microns de largeur et 8,8 microns de profondeur.

Cas #1

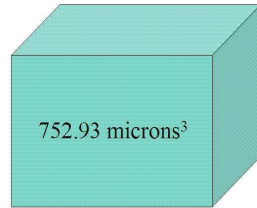


Un voxel

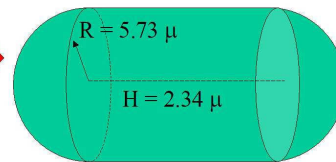


Une sphère ayant la même surface que le voxel

Cas #2

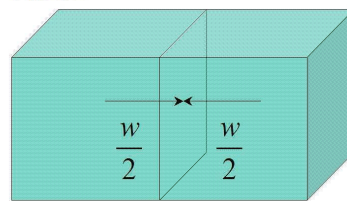


Un voxel

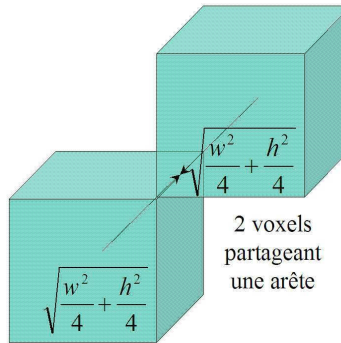


Un tuboïde ayant le même surface extérieure que le voxel

Cas #3

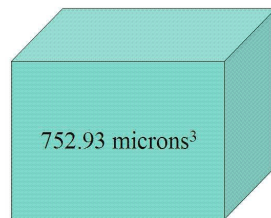


2 voxels partageant une même face

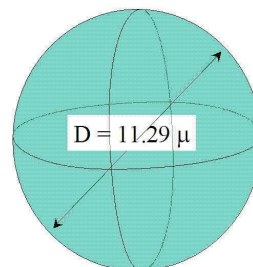


2 voxels partageant une arête

Cas #4

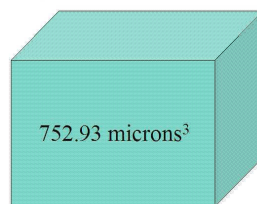


Un voxel

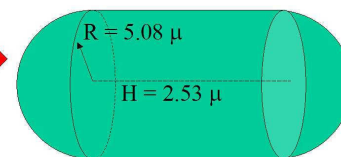


Une sphère ayant le même volume que le voxel

Cas #5



Un voxel



Un tuboïde ayant le même volume que le voxel

Figure 4.5 : Schéma des 5 cas d'étude de la longueur d'un voxel

Ces 5 cas ont été vérifiés avec les données obtenues par le logiciel Unigraphics. Nous avons généré un total de trois géométries avec des valeurs de longueur de la TPB connus afin de pouvoir vérifier l'erreur commise par les différentes façons d'additionner les voxels. La figure 4.6 donne les différentes géométries qui ont été étudiées pour vérifier la directionnalité.

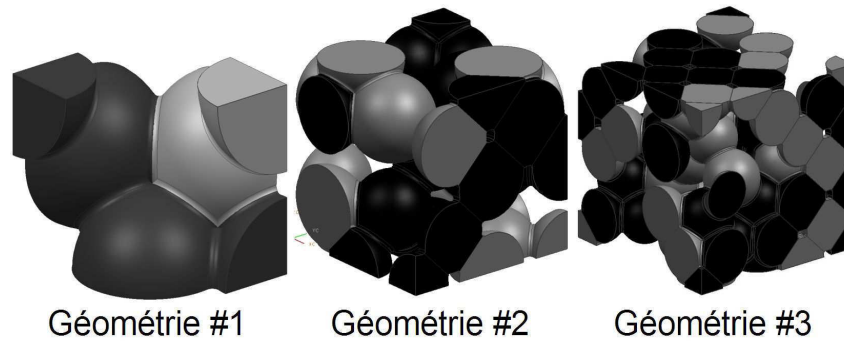


Figure 4.6 : Schéma des 3 géométries utilisées pour l'étude de la directionnalité

Les trois géométries ont des résultats de la longueur de la TPB mesurée de façon continue utilisant le logiciel Unigraphics. Cette mesure a servi de référence pour la vérification de l'application de chaque cas sur les différentes géométries. Le tableau 4.1 illustre les résultats de sensibilité obtenus de la part de chaque géométrie en fonction du nombre de tranches. On présente les résultats des erreurs moyennes sous forme d'histogramme à la figure 4.7. Les erreurs moyennes ont été calculées à partir de la moyenne des 4 valeurs absolues d'écart d'un cas donné pour la géométrie concerné. Par exemple : pour la géométrie #1 cas #1, la moyenne a été appliquée sur le % d'erreur de la 5^{ième} colonne pour ceux concernant la cas #1 seulement (illustrée en jaune dans le tableau 4.1).

Tableau 4.1 : Tableau des résultats de directionnalité et calcul d'erreur

Géométrie Numéro	Nombre tranches	Résultat Cible	Cas 1	% erreur Cas 1	Cas 2	% erreur Cas 2	Cas 3	% erreur Cas 3	Cas 4	% erreur Cas 4	Cas 5	% erreur Cas 5
1	24	1,428	1,619	13,35%	1,543	8,05%	1,228	-13,99%	1,453	1,76%	1,441	0,94%
1	30	1,428	1,617	13,23%	1,489	4,26%	1,343	-5,96%	1,452	1,65%	1,393	-2,43%
2	50	13,380	15,781	17,95%	15,657	17,02%	15,686	17,23%	14,168	5,89%	15,089	12,77%
2	60	13,380	15,711	17,42%	14,595	9,08%	14,780	10,47%	14,105	5,42%	14,129	5,60%
3	75	53,890	59,373	10,17%	58,664	8,86%	58,570	8,68%	53,304	-1,09%	56,163	4,22%
3	90	53,890	59,853	11,06%	61,808	14,69%	60,624	12,50%	53,734	-0,29%	59,149	9,76%
1	100	1,851	1,895	2,38%	1,988	7,41%	1,760	-4,89%	1,701	-8,09%	1,869	0,96%
1	150	1,851	2,005	8,30%	2,407	30,02%	2,197	18,70%	1,800	-2,77%	2,262	22,20%
2	200	13,600	16,424	20,76%	17,751	30,52%	17,427	28,14%	14,745	8,42%	17,133	25,98%
2	300	13,600	16,265	19,59%	18,169	33,59%	17,694	30,10%	14,602	7,37%	17,506	28,72%
3	300	53,890	63,281	17,43%	70,290	30,43%	67,607	25,45%	56,812	5,42%	67,216	24,73%
3	450	53,890	69,573	29,10%	72,743	34,98%	69,847	29,61%	57,850	7,35%	69,480	28,93%

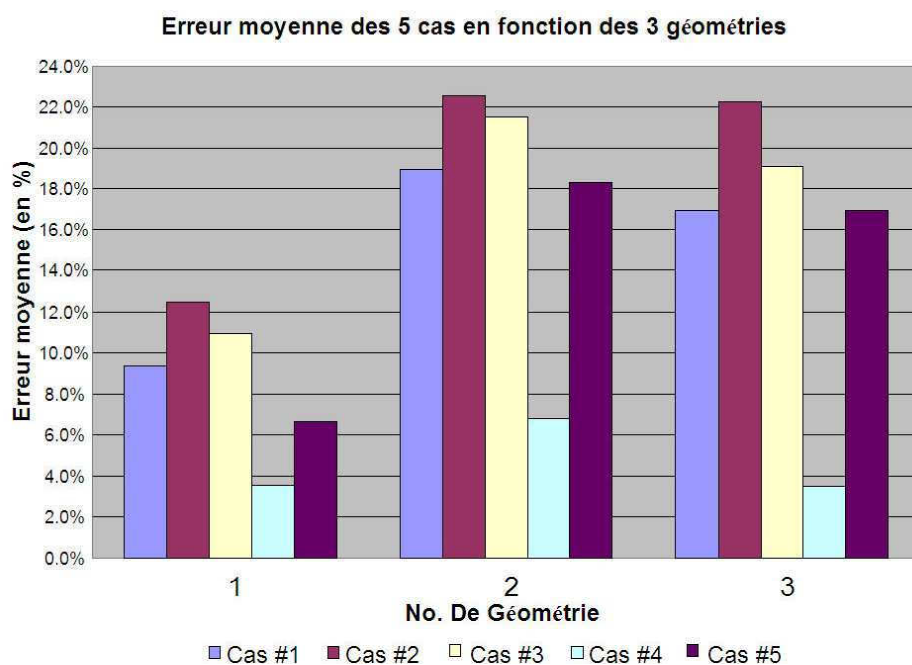


Figure 4.7 : Graphique des écarts moyens en fonction de la géométrie et du cas étudié

De façon globale, on peut noter que le cas #4 présente des résultats de beaucoup meilleurs que tous les autres cas qui sont présentés. Il semble, lorsque la mesure de la longueur caractéristique du voxel est utilisée comme étant le diamètre d'une sphère ayant le même volume que le voxel, les résultats obtenus dans les cas n'ayant pas de distorsions majeures sont meilleurs.

4.6 Algorithme de Hoshen-Kopelman

L'algorithme de Hoshen-Kopelman est un code de calcul qui a été développé dans les années 70. Il s'agit d'un calcul de dénombrement des particules dans un réseau. Il est à la base des calculs relatifs à la percolation. Un des nombreux avantages de cet algorithme est qu'il parcourt le réseau qu'une seule fois grâce à la méthode d'étiquetage astucieuse. Il a été développé par J. Hoshen et R. Kopelman et publié en octobre 1976 (Hoshen et Kopelman 1976) dans le journal Physical Review Letters B. Cet article est à la base des algorithmes de résolution de labyrinthes de 2D. Dans le présent document, nous avons adapté cette méthode d'étiquetage à un réseau en 3D, ce qui constitue un avancement important pour déterminer la conductivité entre un point quelconque de la matrice 3D et la surface électrolytique ou cathodique. La figure 4.8 permet d'expliquer comment la méthode de Hoshen-Kopelman fonctionne.

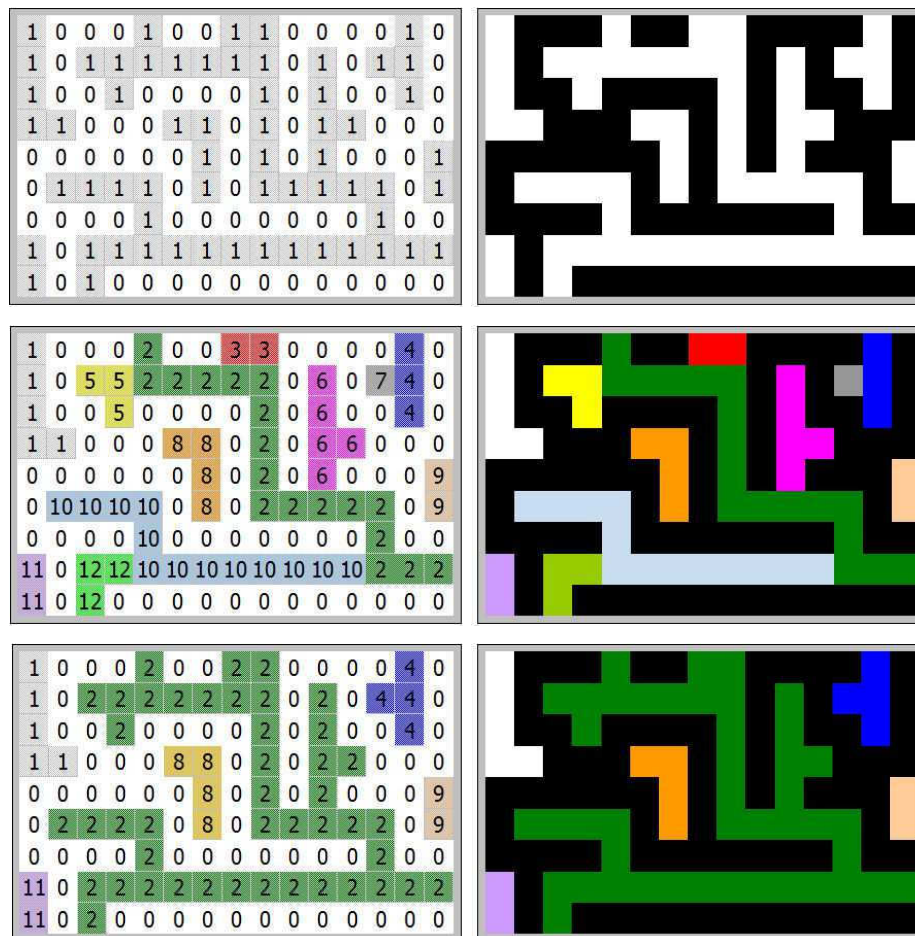


Figure 4.8 : Schéma de principe des 3 étapes de l'algorithme de Hoshen-Kopelman 3D

L'algorithme de Hoshen-Kopelman fonctionne suivant 3 étapes distinctes.

- Étape 1 : Numérotter les sites occupés de la matrice
 - Lors du balayage de la matrice TPB au cours de la première itération, la numérotation des sites se fait par l'attribution d'adresses numériques croissantes à chaque élément qui n'est pas en haut ou à gauche d'un autre élément occupé de la matrice.
- Étape 2 : Égaliser les adresses adjacentes de la matrice
 - Le passage suivant permet d'égaliser voxels par voxels les adresses adjacentes, cette étape se déroule sur plusieurs itérations tant que des voxels adjacents possèdent des adresses différentes.
- Étape 3 : Déterminer la connectivité du voxel dans la matrice
 - Chaque coordonnées de la matrice TPB se voit attribuer une valeur numérique via le résultat de l'algorithme de Hoshen-Kopelman. Si la valeur numérique est identique à celle qui a été attribué sur la face supérieure et la face inférieure, alors la coordonnée est catalogué comme étant connectée.

On peut observer qu'il ne suffit pas de calculer la longueur de la TPB, encore faut-il que les points de contacts formant la jonction de la TPB soient respectivement connectés électriquement avec les surfaces cathodiques et électrolytiques. Pour y arriver, il faut vérifier que les jonctions de la TPB soient actives en résolvant le « labyrinthe » permettant de connecter le point cathodique avec la surface cathodique de l'échantillon et le point électrolytique avec sa surface électrolytique. Chaque point appartenant à la TPB doit être vérifié quant à sa connectivité électrique avec chacune des surfaces respectives (électrolytique et cathodique). On donne, à l'annexe C, le diagramme logique du calcul de la longueur TPB et l'algorithme de Hoshen-Kopelman. L'annexe D présente l'algorithme au complet.

4.7 Analyse de sensibilité

Tout d'abord, nous avons réalisé une analyse de sensibilité qui nous a permis de déterminer l'espacement idéal entre deux coupes. Pour y arriver, nous avons utilisé la tomographie d'un échantillon réel sur 80 tranches, puis nous avons supprimé volontairement des images représentant une coupe tomographique plus épaisse. Par exemple, si nous utilisons les 80

images, l'espacement entre les images étant 10 nm, nous obtiendrons un résultat de 19 677 sites TPB. Or, si nous supprimons les images paires, il ne reste que 40 images et, chaque image est distante de 20 nm, ce qui donne un nouveau résultat de 12 122 sites TPB, et ainsi de suite pour les autres cas d'images retirés. Le tableau 4,2 présente les résultats de cette analyse de sensibilité auxquels jusqu'à 8 images consécutives ont été retirées, ce qui représente un espacement entre les tranches de 90 nm. La figure 4.9 donne le graphique de la longueur de la TPB en fonction du nombre d'images consécutives retirées.

Tableau 4.2 : Résultats de l'analyse de sensibilité

Nombre d'images retirées	Nombre de sites TPB	Grandeur de la matrice	Longueur Caractéristique (nm)	Longueur de TPB (nm/nm ³)
0	19 677	1 383 640	13,5	207,6
1	12 122	691 920	17,4	165,5
2	9 366	467 046	20,6	149,6
3	7 634	345 960	23,4	140,1
4	6 976	276 768	25,9	141,6
5	6 490	247 380	28,1	143,2
6	5 768	207 576	30,2	136,8
7	5 187	172 980	32,2	131,0
8	4 689	155 682	34,0	125,2

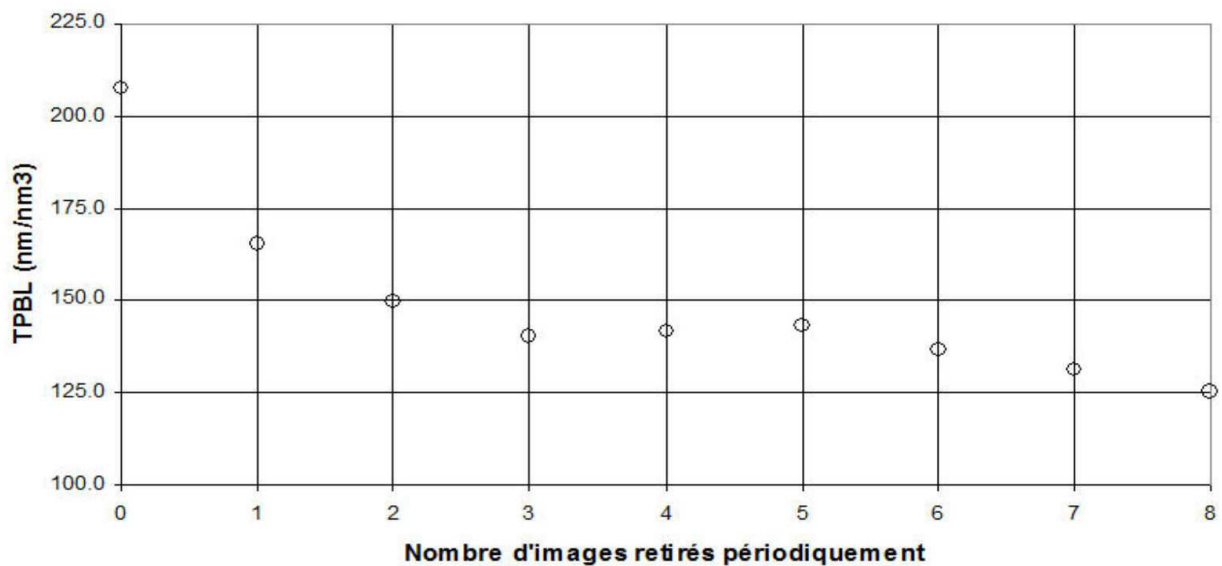


Figure 4.9 : Courbe de sensibilité de la longueur de la TPB en fonction du nombre d'images retirées

Le graphique semble nous donner un plateau entre 3 et 6 images retirées, ce qui signifie que l'espacement idéal entre les tranches pour une tomographie d'un échantillon de cathode GDC-LSM déposé par sérigraphie est d'environ de 30 nm (le point le plus à gauche du plateau). Notons que plus l'espacement entre les coupes est faible, plus la tomographie est longue et plus l'effet rideau peut gêner l'interprétation des images tomographiques produites. En effet, au minimum 4 minutes sont requises pour chaque gravure suivie de la capture d'image. À l'opposé, plus l'espacement entre les coupes est grand, moins précise sera la mesure. Il faut se rappeler que dans l'article de Bajaj paru en 1996, on énonce clairement que pour un artefact aussi petit qu'il soit, le nombre idéal de tranches successives doit être au minimum de 4, afin d'éviter les problèmes de correspondance.

4.8 Courbe de calibration

Une courbe de calibration a été réalisée en utilisant la technique des cathodes à motifs afin de s'assurer une constance dans les périmètres. Les cathodes à motifs ont été testées en utilisant l'analyse par électrode de contact. On présente à la figure 4.10, la microscopie de la surface de la cathode à motifs. L'analyse par électrode de contact a été faite en utilisant un échantillon de 30 cercles cathodiques pour chaque rayon de cathode à motifs. Les rayons qui ont été étudiés sont respectivement de 10 microns, 30 microns, 50 microns et 70 microns en suivant la figure 4.10 de haut en bas.

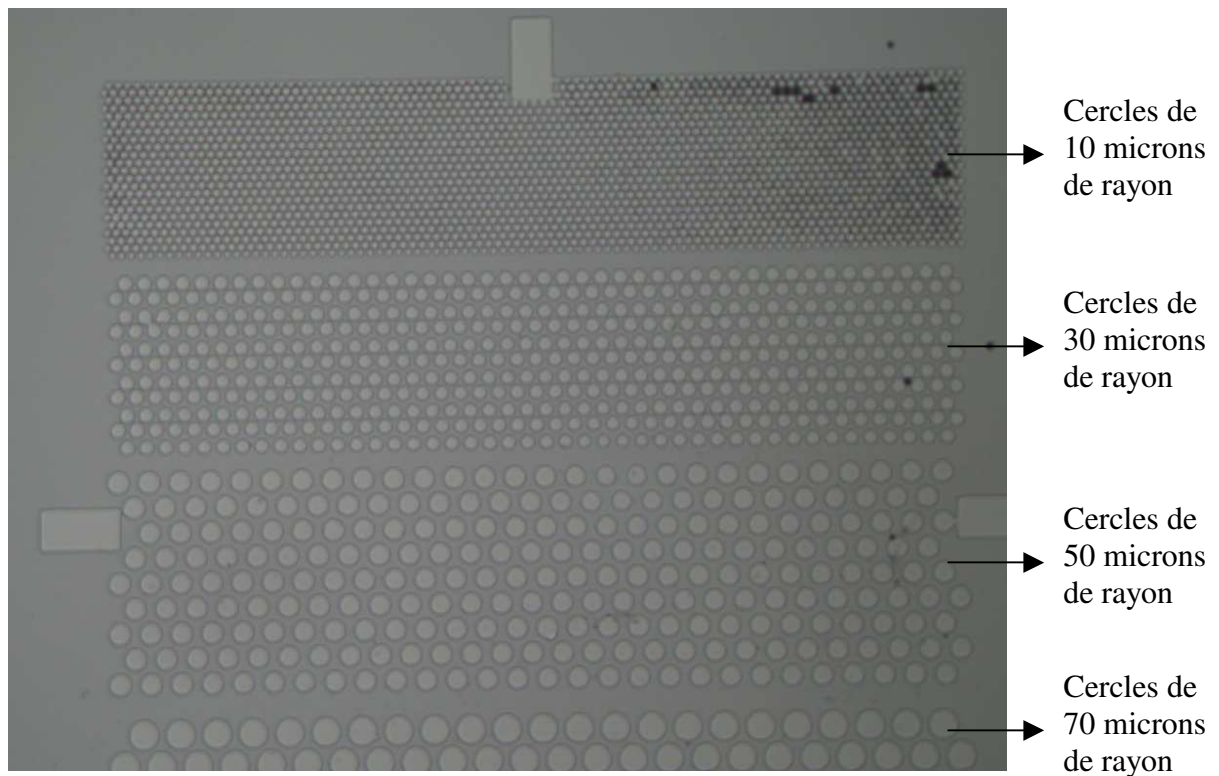


Figure 4.10 : Photo MEB de la cathode à motifs pour la fabrication de la courbe de calibration

Pour chacun des rayons de cathode à motifs, un spectre d'impédance a été réalisé et moyenné sur l'ensemble des 30 échantillons sélectionnés aléatoirement. Pour chaque cathode à motifs, les fréquences ont été balayées de 0,01 Hz à 1 MHz avec une amplitude de 100 mV. Pour chaque test d'impédance, la température a été maintenue à 650°C sous présence d'oxygène. On présente les résultats du spectre d'impédance selon le modèle simplifié de Voigt à la figure 4.12. Une fois les tests réalisés, on a calculé numériquement par lissage la résistance au transfert de charge, puis nous avons fabriqué une courbe de calibration illustrée à la figure 4.13. La courbe de calibration obtenue est représentée sur un graphique semi-log de la résistance en fonction du périmètre. Pour les essais tomographiques d'échantillon réels, un total de deux essais a été exécuté pour déterminer la mesure de la porosité, la mesure de la longueur de la TPB et le pourcentage des TPB connectés électriquement.

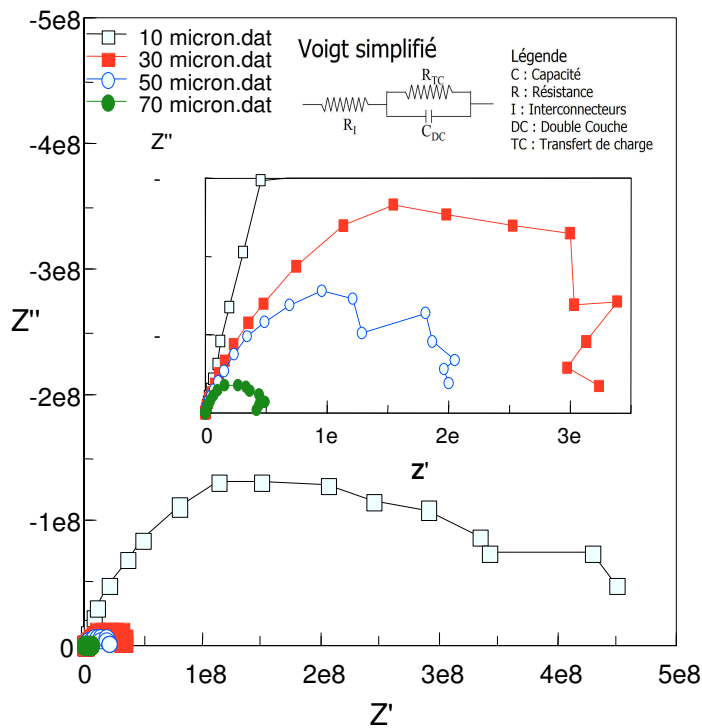


Figure 4.11 : Spectroscopie d'impédance des électrodes à motifs.

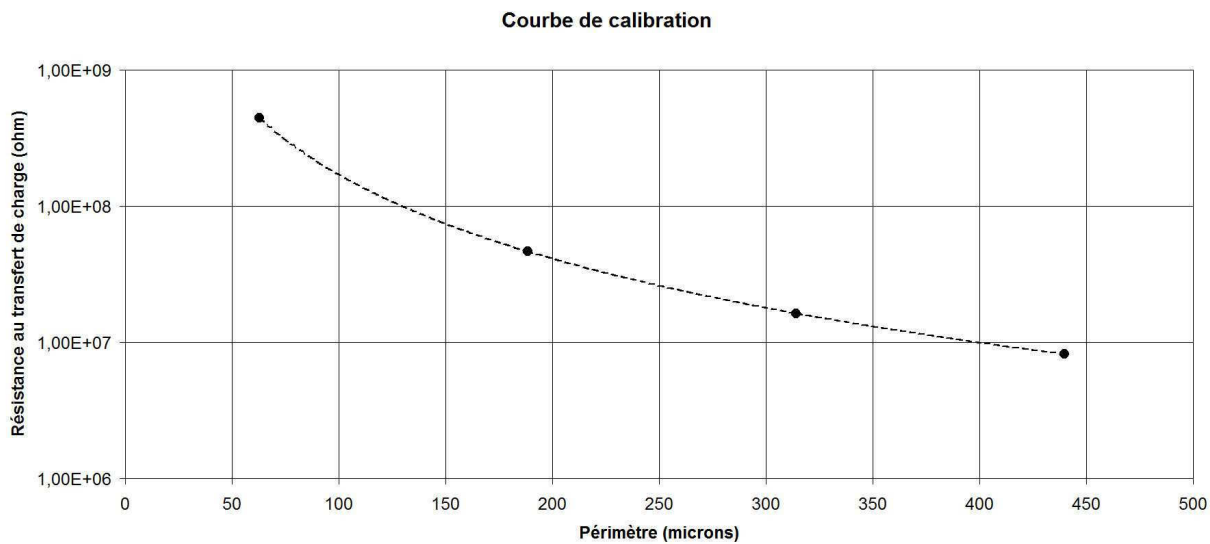


Figure 4.12 : Courbe de calibration de la résistance au transfert de charge en fonction du périmètre

Un premier essai a été produit en utilisant un total de 18 images espacées de 10 nm. La porosité obtenue était de 45,8% pour une longueur de la TPB de 86,3 nm / nm³. Dans ce test, le nombre

d'images étant trop faible, il était impossible de calculer le pourcentage de la TPB qui est connectée. Cette réalité a nécessité un autre essai avec un nombre d'images beaucoup plus grand. Dans le but de maintenir la longueur caractéristique identique, nous avons conservé les paramètres de capture pour le nouvel échantillon à l'étude. Les résultats de l'essai suivant ont permis d'avoir toutes les mesures requises avec un échantillon de 80 images espacées de 10 nm. La porosité obtenue était de 42,8% pour une longueur de la TPB de 207,6 nm / nm³. Le nombre d'images étant suffisamment grand, l'algorithme de Hoshen-Kopelman a donné un pourcentage de la TPB connecté de 4,2% seulement. Cette valeur plutôt faible est due à la composition de notre échantillon fait de 50% massique de LSM et 50% massique de GDC qui a été utilisé comme autre constituant de la cathode. Comme le LSM a une masse molaire plus faible que celle du GDC, le pourcentage volumique du LSM est bien plus élevé. Cet essai, utilisant plus de tranches par unité de volume, offre une meilleure résolution et par conséquent il est plus précis que le précédent. Ce résultat permet de faire une interpolation dans la courbe de calibration en fonction de la longueur de la TPB connectée. En effet, la longueur de la TPB connectée étant de 207,6 nm multipliée par le pourcentage de connectivité de 4,2%, on obtient une longueur de la TPB connectée de 8,72 nm pour un échantillon de 1 nm³. Supposons un échantillon de cathode poreuse formée de LSM et de GDC sérigraphiée ayant une dimension de 30 nm × 30 nm × 30 nm, on constate que le volume calculé sera de 27000 nm³. Par conséquent, on aura alors une longueur de la TPB connectée attendue, selon les résultats obtenus de 235,4 microns. Ce résultat numérique, une fois interpolé dans la courbe de calibration, permet de conclure que la résistance au transfert de charge de cet échantillon serait de $2,712 \times 10^7$ ohm.

CHAPITRE 5

Conclusion

5.1 Sommaire

Dans le présent travail de recherche, la cathode analysée, constituée de GDC et de LSM, avait une conduction électronique plus prépondérante que la conduction ionique. Selon plusieurs auteurs, une cathode non ionique possède une densité de courant proportionnelle à la longueur de la TPB (Vielstich, Lamm et coll. 2003; Koep, Compson et coll. 2005). La difficulté principale de ce projet de recherche était de pouvoir mesurer la longueur de la TPB, particulièrement sa connectivité. Pour ce faire, un protocole faisant appel à la tomographie par FIB a été développé pour analyser une cathode poreuse. Ce protocole fait intervenir une technique d'enrobage des échantillons qui permet de faire ressortir les différentes phases constituantes d'une cathode. Cette technique d'enrobage plus pénétrante utilise une résine biologique qui réagit suivant une réaction catalytique avec le cérium. Une fois l'échantillon de la cathode enrobé, la prochaine étape consiste à procéder à la tomographie par coupes successives. La tomographie par FIB est alternée avec une capture via la MEB afin de réaliser une série d'images espacées de 10 nm chacune.

La haute résolution de la technique de mesure nécessitait l'utilisation de particules nanométriques afin de créer un échantillon de cathode qui puisse être analysé. En conséquence, le recours aux cathodes témoins requérait que l'on utilise la technique de sérigraphie de poudres nanométriques de LSM et de GDC dans le but de créer des cathodes poreuses nanostructurées.

Une fois la tomographie réalisée, les tranches obtenues ont été soumises à un traitement d'images permettant de distinguer les différentes phases constituantes de la cathode; c'est-à-dire celle du matériel cathodique (LSM), celle du matériel électrolytique (GDC) et la porosité (volume vide interstitiel). Ces matrices 3D ont été le principal intrant du programme Matlab qui fut construit pour mesurer la longueur de la TPB. Le programme Matlab a subi des ajouts, notamment le calcul de la porosité ainsi que du pourcentage de connectivité provenant de la compilation de l'algorithme de Hoshen-Kopelman, adapté en 3D aux fins de cette recherche.

Par la suite, des vérifications portant sur l'exactitude du résultat obtenu ont été menées. Les essais pilotes furent effectués à l'aide de structures simples. Ces derniers ont montré que certains voxels durant la compilation étaient comptés en double. Ce problème fut corrigé en mettant au point un filtre numérique.

Pour estimer l'espacement idéal entre deux coupes tomographiques, des tests de sensibilité sur une structure cathodique déposée par sérigraphie furent menés. Aussi, puisque l'unité du résultat de la mesure est en voxels / voxels³, il a fallu convertir cette mesure en unités de mesure de volume. Un logiciel appelé Unigraphics a permis de générer trois géométries aléatoires basées sur des particules sphériques dont la longueur de la TPB était préalablement connue. Il fut démontré que lorsque la structure n'a pas de distorsion majeure, c'est-à-dire que les voxels ne dévient pas significativement de la forme cubique, il est possible de calculer la longueur caractéristique comme étant le diamètre d'une sphère ayant le même volume que le voxel.

Suite aux ajustements requis à l'algorithme, le code a été appliqué avec succès sur une structure réelle de 80 images obtenues par la tomographie. Les résultats démontrent que pour une cathode constituée de LSM et de GDC déposés par sérigraphie, la longueur de la TPB mesurée est de 207,6 nm / nm³. L'algorithme a permis de mesurer une porosité de 42,8% ainsi qu'un pourcentage de connectivité de seulement 4,2%.

Une courbe de calibration fut également créée afin de traduire le résultat de la longueur de la TPB réelle à une mesure électrochimique de résistance au transfert de charge. La courbe de calibration a été réalisée à partir de cathodes à motifs déposés par photolithographie. Quatre périmètres ont été testés via un système d'analyse par électrode de contact. La courbe de calibration a permis de déterminer la résistance au transfert de charge correspondante à un exemple précis d'une cathode ayant une taille de 30 nm x 30 nm x 30 nm qui selon les résultats obtenus aurait une résistance au transfert de charge de $2,712 \times 10^7$ ohm.

5.2 Contributions

Ce travail de recherche portant sur les piles à combustible SOFC contient plusieurs contributions.

Premièrement, le développement d'un outil de mesure de la TPB à partir d'une cathode composée de LSM et de GDC. Notons que cet outil de mesure peut également être applicable à n'importe quel matériau poreux composé de deux phases ayant des masses molaires différentes.

Deuxièmement, la réalisation d'une tomographie de haute résolution atteignant 250 nm^3 / voxel. Pour ce faire, il a fallu créer des poudres par la méthode GNP afin de constituer une cathode ayant des structures fines. Cette réalisation a été possible via l'utilisation de la résine biologique qui nous a permis de faire une imprégnation très pénétrante.

Troisièmement, la création d'un protocole pouvant transformer les images obtenues par tomographie en matrices binaires 3D. Cette étape a permis de séparer numériquement les constituants de la cathode via le traitement d'images.

Finalement, l'adaptation d'un algorithme de Hoshen-Kopelman en 3D, ce qui a permis de vérifier la connexion électrique de la TPB avec leurs surfaces respectives. L'algorithme permet alors de calculer la connectivité des voxels constituant la matrice de la TPB.

Au cours du développement de l'algorithme, une série de tests fut réalisée pour s'assurer que le résultat obtenu provenant du code Matlab était celui reflétant la réalité. Pour ce faire, des tests de sensibilité ont été menés ainsi que des tests de directionnalité. Ils ont permis de maximiser la fiabilité et la signification statistique des valeurs de longueur de la TPB estimée.

5.3 Travaux futurs

Une suite logique aux travaux découlant de ce doctorat serait de faire des ajouts à l'algorithme de calcul permettant de mesurer également la tortuosité de surface, la porosité ouverte et fermée. Ces modifications permettraient de profiter au maximum des informations reçues par

la tomographie. Elles permettraient également de rendre l'algorithme de calcul beaucoup plus performant.

La tomographie proposée ne pouvait être réalisée que sur certains matériaux seulement en raison de limitations associées à l'utilisation des équipements. Il serait souhaitable d'étendre cette analyse à d'autres matériaux cathodiques ou anodiques. Dans le présent travail, il a fallu approuver préalablement chaque matériau à analyser, ce qui limitait beaucoup les combinaisons de matériau cathodiques possibles.

Les prochaines étapes doivent inclure l'établissement d'un lien entre la longueur de la TPB réelle obtenue par tomographie et les résultats électrochimiques d'une pile complète utilisant la même combinaison de matériau et le même protocole de fabrication de la cathode du présent travail.

En raison du fait que le travail de polymérisation a été fait en dehors du cadre de ce travail, il serait opportun d'approfondir le protocole d'imprégnation afin d'en améliorer son efficacité. La résine LR White est principalement utilisée à des fins biologiques. Pour pouvoir l'utiliser dans un but tomographique, il est nécessaire de changer sa constitution chimique afin de rehausser son pouvoir pénétrant ainsi que la capacité à distinguer les phases de l'échantillon à analyser lors d'une analyse MEB. Dans le contexte d'un futur projet, il faudrait examiner la possibilité de pouvoir synthétiser notre propre résine d'imprégnation afin d'augmenter l'efficacité de cette étape cruciale.

Finalement, soulignons que le développement de cet outil de mesure de la TPB peut être appliqué à n'importe quelle combinaison de matériaux poreux ayant une masse molaire différente. Une extension de ce projet de recherche serait de pouvoir appliquer ce protocole de mesure à d'autres domaines pour lesquels la longueur d'une TPB est importante, par exemple, dans le domaine des catalyseurs, dans le domaine des alliages et dans le domaine biomédical.

LISTE DES RÉFÉRENCES

- Abbaspour, A., K. Nandakumar, et coll. (2006). "A novel approach to study the structure versus performance relationship of SOFC electrodes." Journal of Power Sources **161**(2): 965-970.
- Bajaj, C. L., E. J. Coyle, et coll. (1996). "Arbitrary topology shape reconstruction from planar cross sections." Graphical Models and Image Processing **58**(6): 524-543.
- Betoule, O. (1994). Influence des distributions de vitesse et température de jet de plasma d'arc et de particules sur les propriétés des dépôts d'alumine. Sciences des matériaux. Limoges, Université de Limoges: 165 pages.
- Bieberle, A. et L. J. Gauckler (2000). "Reaction mechanism of Ni pattern anodes for solid oxide fuel cells." Solid State Ionics **135**(1-4): 337-345.
- Brichzin, V., J. Fleig, et coll. (2002). "The geometry dependence of the polarization resistance of Sr-doped LaMnO₃ microelectrodes on yttria-stabilized zirconia." Solid State Ionics **152-153**: 499-507.
- Chavan, S. V. et A. K. Tyagi (2004). "Preparation and characterization of Sr_{0,09}Ce_{0,91}O_{1,91}, SrCeO₃, and Sr₂CeO₄ by glycine-nitrate combustion: Crucial role of oxidant-to-fuel ratio." Journal of Materials Research **19**(11): 3181-3188.
- Davies, S. et B. Khamsehpour (1996). "Focused ion beam machining and deposition for nanofabrication." Vacuum **47**(5): 455-462.
- de Boer, B., M. Gonzalez, et coll. (2000). "The effect of the presence of fine YSZ particles on the performance of porous nickel electrodes." Solid State Ionics **127**(3-4): 269-276.
- Gostovic, D., J. R. Smith, et coll. (2007). "Three-dimensional reconstruction of porous LSCF cathodes." Electrochemical and Solid-State Letters **10**(12): B214-B217.
- Holzer, L., F. Indutnyi, et coll. (2004). "Three-dimensional analysis of porous BaTiO₃ ceramics using FIB nanotomography." Journal of Microscopy **216**(1): 84-95.
- Holzer, L., B. Muench, et coll. (2006). "FIB-Nanotomography of Particulate Systems ; Part I: Particle Shape and Topology of Interfaces." Journal of the American Ceramic Society **89**(8): 2577-2585.
- Hoshen, J. et R. Kopelman (1976). "Percolation and cluster distribution. I. Cluster multiple labelling technique and critical concentration algorithm." Physical Review B (Solid State) **14**(8): 3438-45.

- Huang, K., M. Schroeder, et coll. (1999). "Oxygen permeation through composite oxide-ion and electronic conductors." Electrochemical and Solid-State Letters **2**(8): 375-378.
- Ishihara, T., S. Fukui, et coll. (2002). "Mixed electronic-oxide ionic conductor of BaCoO₃ doped with La for cathode of intermediate-temperature-operating solid oxide fuel cell." Solid State Ionics **152-153**: 609-613.
- Kenney, B. et K. Karan (2010). "Estimation of Chemical and Transport Processes in Porous, Stoichiometric LSM Cathodes Using Steady-state Polarization and Impedance Modeling." Journal of the Electrochemical Society **157**(8): 1126-37.
- Koep, E., C. Compson, et coll. (2005). "A photolithographic process for investigation of electrode reaction sites in solid oxide fuel cells." Solid State Ionics **176**(1-2): 1-8.
- Kubis, A. J., G. J. Shiflet, et coll. (2004). "Focused ion-beam tomography." Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science **35 A**(7): 1935-1943.
- Lasagni, F., Lasagni, A., et coll. (2008). "Nano-characterization of Cast Structures by FIB-Tomography." Advanced Engineering Materials **10**(1-2): 62-66.
- Lorensen, W. E. et H. E. Cline (1987). "Marching Cubes: A High Resolution 3D Surface Construction Algorithm." SIGGRAPH '87 Conf Proc, July 27, 1987 - July 31, 1987 **21**(4): 163-169.
- Lvovich, V. F. (2012). Impedance Spectroscopy: Applications to Electrochemical and Dielectric Phenomena. New York, Wiley.
- Matsuzaki, Y. et I. Yasuda (2002). "Electrochemical properties of reduced-temperature SOFCs with mixed ionic-electronic conductors in electrodes and/or interlayers." Solid State Ionics **152-153**: 463-468.
- Metcalfé, C., O. Kesler, et coll. (2009). Three-phase boundary length evaluation in modeled sintered composite solid oxide fuel cell electrodes. 11th International Symposium on Solid Oxide Fuel Cells (SOFC-XI)- 216th ECS Meeting, October 4, 2009 - October 9, 2009, Vienna, Austria, Electrochemical Society Inc.
- Mizusaki, J., H. Tagawa, et coll. (1994). "Preparation of nickel pattern electrodes on YSZ and their electrochemical properties in H₂-H₂O atmospheres." Journal of the Electrochemical Society **141**(8): 2129-2134.

- Mokkelbost, T., I. Kaus, et coll. (2004). "Combustion synthesis and characterization of nanocrystalline CeO₂-based powders." Chemistry of Materials **16**(25): 5489-94.
- Peng, R., C. Xia, et coll. (2002). "Sintering and electrical properties of (CeO₂)_{0,8}(Sm₂O₃)_{0,1} powders prepared by glycine-nitrate process." Materials Letters **56**(6): 1043-1047.
- Philibert, J. (1985). Diffusion et transport de matière dans les solides. Cedex, France, Les Ulis.
- Radhakrishnan, R., A. V. Virkar, et coll. (2005). "Estimation of Charge-Transfer Resistivity of La_{0,8}Sr_{0,2}MnO₃ Cathode on Y_{0,16}Zr_{0,84}O₂ Electrolyte Using Patterned Electrodes." Journal of The Electrochemical Society **152**(1): A210-A218.
- Radhakrishnan, R., A. V. Virkar, et coll. (2005). "Estimation of Charge-Transfer Resistivity of Pt Cathode on YSZ Electrolyte Using Patterned Electrodes." Journal of The Electrochemical Society **152**(5): A927-A936.
- Rivard, T., F. Gitzhofer, et coll. (2012). "Geometrical triple phase boundary length measurement using focused ion beam tomography." Canadian Journal of Chemical Engineering **90**(3): 712-718.
- Srivastava, P. K., T. Quach, et coll. (1997). "Electrode supported solid oxide fuel cells: Electrolyte films prepared by DC magnetron sputtering." Solid State Ionics **99**(3-4): 311-319.
- Vielstich, W., A. Lamm, et coll. (2003). Handbook of Fuel Cell. New York, Wiley.
- Virkar, A. V., J. Chen, et coll. (2000). "The role of electrode microstructure on activation and concentration polarizations in solid oxide fuel cells." Solid State Ionics **131**(1-2): 189-198.
- Wang, S., Y. Jiang, et coll. (1998). "Promoting effect of YSZ on the electrochemical performance of YSZ+LSM composite electrodes." Solid State Ionics **113-115**: 291-303.
- Wilson, J. R., W. Kobsiriphat, et coll. (2006). "Three-dimensional reconstruction of a solid-oxide fuel-cell anode." Nature Materials **5**(7): 541-544.

ANNEXE A

Article du Canadian Journal of Chemical Engineering

GEOMETRICAL TRIPLE PHASE BOUNDARY LENGTH MEASUREMENT USING FOCUSED ION BEAM TOMOGRAPHY

Tony Rivard, François Gitzhofer* and Nicolas Abatzoglou*

Department of Chemical Engineering, Université de Sherbrooke, Sherbrooke, Quebec, Canada J1K 2R1

In multi-component materials, triple phase boundary (TPB) is the location where reactions occur. A typical example is the TPB encountered in solid oxide fuel cells at the cathode–electrolyte interface. We proposed a tomographic approach that was developed based on serial sectioning using a focused dual ion beam (FIB) system. For image capture, FIB tomography was coupled with scanning electron microscopy, and differentiation of the composite cathode materials was possible through image contrast adjustment. An algorithm, built on the Hoshen–Kopelman theory, was then applied to measure TPB length. The percentage of the connected TPB line was also calculated with the algorithm for 3D computation.

Keywords: focused ion beam, serial sectioning, SOFC, tomography, triple phase boundary length measurement

INTRODUCTION

Recent advances in microscopy allow us to obtain information about nanometric-scale materials. In their application to fuel cell technology, various microscopy techniques, such as high-resolution microscopy, performed on fuel cell electrodes, have generated important data on electrode porosity, electrode composition and material distribution (Tsoga et al., 1999). In solid oxide fuel cell (SOFC) devices with non-ionic conducting cathodes, electrochemical reactions mostly occur at the triple phase boundary (TPB) layer. The development of SOFC current density prediction models, therefore, depends on our ability to quantify TPB length (TPBL) and, furthermore, to determine connected TPBL. In three-dimensional (3D) geometry, a TPB is considered to be connected if an ion path is electrically linked to both the electrode and electrolyte surfaces. Identification and measurement of material characteristics on the nanometric scale are required for accurate, connected TPBL estimation.

This work presents the development of a characterisation method based on the serial sectioning of a studied structure, the use of numerical algorithms and interpretation of the results. The overall objective is to estimate TPBL and the connectivity of an electrode to predict current density according to a tomographic approach. The first part of our experiment involves the computation of image stacks with the Hoshen–Kopelman algorithm to calculate connected TPBL in voxels/voxels³ and the ratio of connected TPBL. The second part is dedicated to sensitivity analysis of

the method and comments on the assumptions made in assessing TPBL in nm/nm³.

LITERATURE REVIEW

Serial sectioning is widespread in material science. Many authors have reported etching techniques with focused ion beams (FIBs) in material microstructure analysis. A key article (Holzer et al., 2004), has presented the results of BaTiO₃ 3D study by FIB nanotomography. Its authors adopted a serial-sectioning procedure for the 3D assessments of a 1-phase material impregnated with epoxy resin. Voxel resolution by nanotomography was as low as 714 nm³/voxel. Porosity was calculated by 3D pore space analysis with both laser granulometry and FIB nanotomography in comparison to mercury intrusion porosimetry (MIP). Significant differences were observed between MIP and 3D pore space data. The authors attributed these differences to the inkbottle effect that led to the overestimation of small pore frequency. Also, FIB

* Author to whom correspondence may be addressed.

E-mail addresses: francois.gitzhofer@usherbrooke.ca,

nicolas.abatzoglou@usherbrooke.ca

Can. J. Chem. Eng. 90:712–718, 2012

© 2011 Canadian Society for Chemical Engineering

DOI 10.1002/cjce.20570

Published online 20 July 2011 in Wiley Online Library (wileyonlinelibrary.com).

nanotomography provided quantitative information on smaller than 100-nm size closed pores that could not be achieved with other methods.

The same research group (Holzer et al., 2006) published another important article on the 3D nanotomography of particulate systems. Their work demonstrated that it was possible to identify individual grains, even of complex structures. Five different grain size fractions of tested cement were impregnated in Wood's metal, a fusible alloy of bismuth that has a melting point of 70°C. The samples were sputtered with platinum to prevent mechanical drift during the FIB slicing procedure. Voxel resolution was 1560 nm³/voxel at 35 000× magnification. They also reported that drift correction was very influential for accurate 3D data when performing high-resolution nanotomography.

Schneider et al. (2006) developed discrete electrode modelling based on spherical, monosized and randomly positioned particles. They focused on how electrode operation depended on electrode composition, temperature and thickness. Their model was built on the resistance network theory and took into account the sintering impact of particles on TPB results.

Abbaspour et al. (2006) produced a 2D computer model for the TPB measurement of lanthanum strontium manganate–yttria-stabilised zirconia (LSM–YSZ) cathodes. They focused on the effect of cathode composition on various electrode properties, such as electrode surface area and polarisation resistance. Their model showed good agreement with other experimental and simulation data. Nevertheless, these authors specifically mentioned their difficulty in taking the third dimension into account during numerical model development.

In the same year, Wilson et al. (2006) elaborated a serial-sectioning procedure with dual-beam FIB for the 3D reconstruction of anode structures. Sections 50-nm thick were removed from the exposed surface, followed by scanning electron microscopy (SEM) with a total analysed sample volume of 105 μm³.

More recently, in 2007, another research group, headed by Gostovic et al. (2007), studied the electrochemically active region of a lanthanum strontium cobalt ferrite cathode reconstructed in 3D with dual beam FIB. The reconstructed volume totalled 1065 μm³ from the free air surface to the dense yttria-stabilised zirconia electrolyte interface. They measured various microstructural properties, such as overall porosity, closed porosity, graded porosity, surface area, tortuosity, TPBL and pore size. Electrochemical impedance spectroscopy and microstructure data were correlated.

IMAGING THEORY

Definition of Voxels

A 'voxel' is a volume element representing an intensity value on a regular grid in 3D space. As a pixel defines a point in 2D space by means of *x* and *y* coordinates, a third coordinate is needed in 3D space. A voxel is described in terms of its position, colour, and density in 3D space. In this work, TPBL is embodied by the dimensionless number defined by Equation (1). It is the ratio of the total number of voxels belonging to and representing the TPB over the sum of all voxels in the 3D matrix generated by FIB nanotomography (the sample stack):

$$\text{TPBL} = \frac{\sum_{\text{TPB}} (\text{Voxel})}{\sum (\text{Total voxels})} \quad (1)$$

Total sample volume

Characteristic Voxel Length

While SEM coupled with FIB can determine characteristic length, TPBL defined by Equation (1) can be converted to a more convenient unit length. Characteristic voxel length corresponds to the diameter of a sphere having an outer surface equivalent to that of voxels. In the present work, voxel height, width and depth values corresponded to 9.8, 9.8 and 10.0 nm, respectively, for the first run, and 5.0, 5.0 and 10.0 nm, respectively, for the second run, with the Dual Zeiss 1540XB Crossbeam FIB used for tomography. As an example, two adjacent voxels, in whatever direction, have a centre-to-centre distance equal to two characteristic lengths. With this voxel characteristic length definition, characteristic length is equal to 13.5 nm for the first run and 8.6 nm for the second run for previously given voxel size. Characteristic length also serves to express TPBL, from TPB values, to a more usual unit scale, such as 'nm of TPBL/nm³'.

Hoshen–Kopelman Algorithm

The Hoshen–Kopelman algorithm was developed to determine critical percolation concentration, percolation probabilities and cluster size distribution, as employed in filtration processes (Hoshen and Kopelman, 1976). The present work proposes a modified Hoshen–Kopelman algorithm for electrical conduction criteria between the TPB voxel and the surface represented by the electrode or the electrolyte. Figure 1 provides a complete detailed scheme of the algorithm for an easier understanding. The algorithm must be applied on 1-bit matrices containing binary values that could be either 1s or 0s, meaning either occupied or unoccupied sites, in 3D matrices. Figure 2a gives an example of a 1-bit matrix represented by occupied and unoccupied sites. This algorithm, which has two major iterations, was applied to 1-bit matrices obtained from the FIB image stack. After construction of the 1-bit matrix with both occupied and unoccupied sites, the next step (first iteration of the algorithm) consists of labelling each occupied voxel with a numerical address, as shown in Figure 2b. The imposed constraint is that two adjacent voxels in a matrix bear the same numerical address. The next step (second iteration of the algorithm) is equalisation of the addresses with bits having the lowest neighbouring numerical address. In practice, these computing steps mean that the algorithm defines the pathways of electrical conduction (i.e. contact between particles) through the matrix. Finally, if the address corresponding to the TPB coordinates is equal to an address on both the electrolyte and the electrode phase, the corresponding TPB is considered to be connected, as illustrated in Figure 2c. More details on this testing procedure are presented in the Discussion Section.

EXPERIMENTAL PROTOCOL

Cathode Sample Preparation

Screen-printing

SOFC cathode samples were prepared by screen-printing on a substrate (Peng et al., 2002a; 2004; Bouchard et al., 2005). The cathode was composed of 50 wt.% gadolinium-doped ceria (GDC) and 50 wt.% LSM. GDC and LSM powders were synthesised by the glycine–nitrate procedure (Peng et al., 2002b; Ji et al., 2003; Chavan and Tyagi, 2004). The powders were mixed with 10 wt.% carbon black as pore precursor. A few drops of alpha-Terpineol were added for the screen-printing process and mixed until a viscous cathode paste was produced. The paste was spread on the

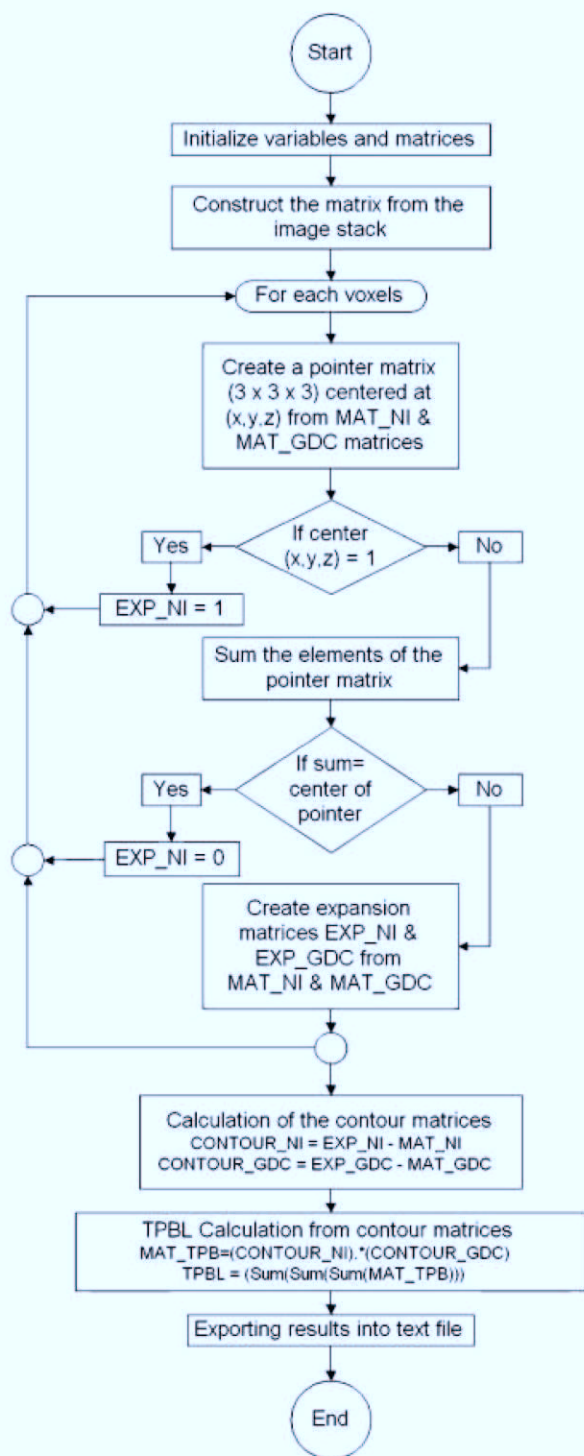


Figure 1. Scheme of the Hoshen–Kopelman algorithm.

screen with a scraping blade, after which the screen was removed. On completion of screen-printing, the cathode substrates were fired at 1200 °C for about 2 h. Carbon black was then ‘burnt off’, leaving a nanoporous matrix for TPBL measurement by the FIB serial-sectioning procedure.

Embedding

Samples produced by screen-printing were impregnated in LR white resin from Electron Microscopy Science (Hatfield, PA), and

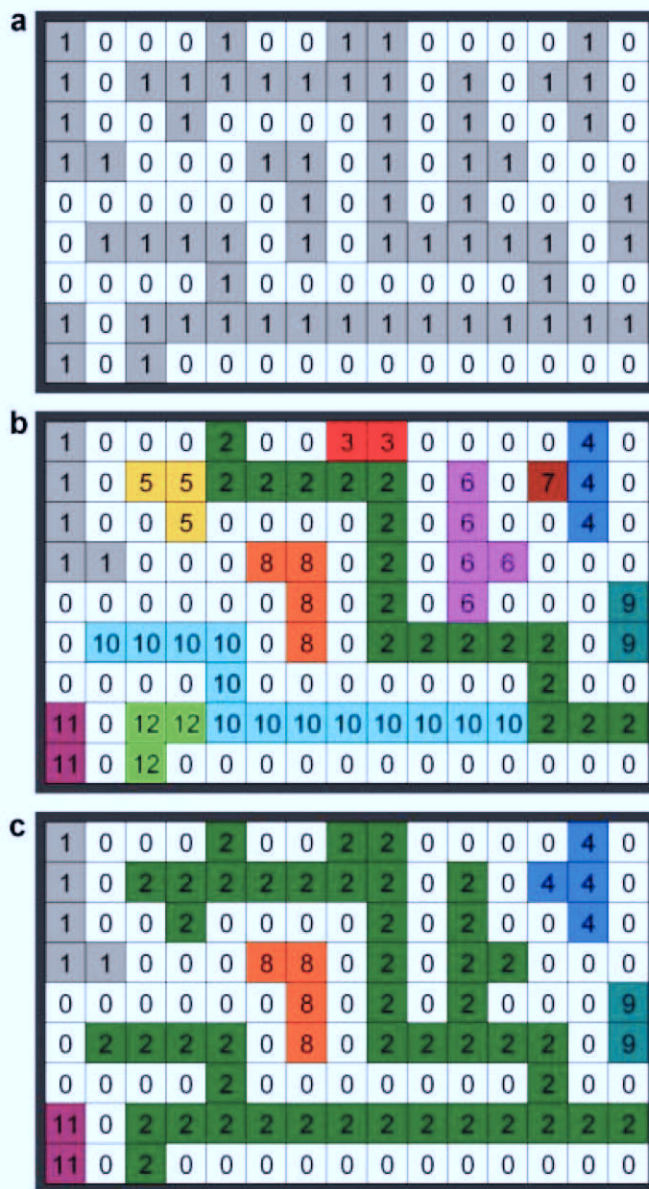


Figure 2. Application example of the Hoshen–Kopelman algorithm. a: original matrix, b: first iteration (labeling pass), c: last iteration (connection pass)

modified according to the protocol described below. The resin, needed to fill the nanopores of the cathode sample, was chosen because of its highly porous, material-penetrating properties. A special procedure for resin polymerisation was developed to ensure that impregnation was successful. The procedure consists of initiating a cross-linking reaction into the LR white resin by adding polystyrene before onset of the polymerisation reaction. The cross-linking reaction was catalysed by cerium, a cross-linking agent present in the sample. This embedding step was crucial to differentiate between each phase of the cathode material and, consequently, to measure real TPBL. According to the method, the sample to be embedded is immersed in 10 mL of LR white resin containing 20 mg of small polystyrene strips. The LR white initiator is then added, and the sample is placed in a desiccator at about 60 °C under vacuum. Polymerisation lasts for at least 48 h before reaching full conversion. At this point, the viscous phase in the impregnated sample is completely solidified. The

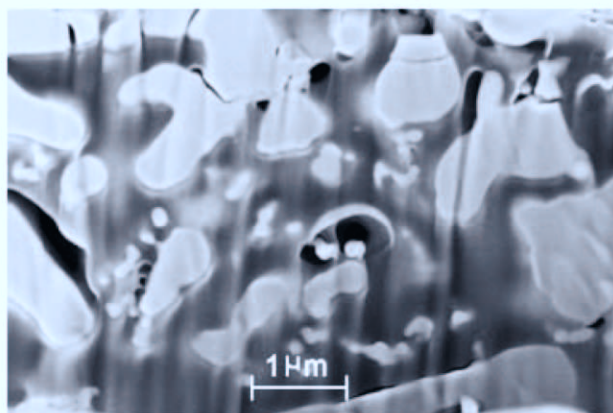


Figure 3. FIB image of a sample with no cross-linking reaction.

sample should have a whitish colour and must be very opaque. It is then broken in half and finely polished with a polishing disk to ensure that the cathode surface is sufficiently exposed. The solidified resin is mechanically removed entirely to prevent drift problems during the subsequent FIB process. After resin removal, the cathode sample is 'sputter-coated' with low-pressure gold-palladium plasma prior to FIB nanotomography. This procedure works well for any materials (biological, medical and nanomaterials) to be embedded, but it has to be noted that if no cerium (a cross-linking agent) is present in the sample, a trace of it or any other cross-linking agent that does not harm the sample should be added to the reaction mixture, to ensure and enhance the necessary cross-linking reaction. If LR white resin does not make cross-links, the resin will not enhance the desired optical properties. Figure 3 is a micrograph of a sample with no polystyrene added to the reaction mixture; in this case, no cross-linking reaction occurs, and it is impossible to clearly differentiate each of the material's phases during subsequent steps.

3D Reconstruction

FIB nanotomography

FIB nanotomography is an etching process performed with a Dual Zeiss 1540XB Crossbeam workstation deploying a gallium ion beam that has a cutting probe less than 7.0 nm in diameter. SEM was undertaken with an InLens detector at 3-kV acceleration voltage. The cathode was fixed in the sample holder with aluminium tape for electrical grounding. It was initially examined under low magnification to locate the most appropriate etching zone that had to be exempt from major imperfections, as large porosities or cracks could interfere with SEM image capture. For tomography, etching was alternated with backscattered SEM image capture at the same 'selected zone' on the sample. Average thickness of the slices was set to 10.0 ± 0.1 nm to ensure best image quality. All the desired images 'captured' were then processed by commercial JASC Paintshop Pro software for contrast adjustment/image clarification. Two images were obtained from each tomographic section, one for each solid component of the cathode. Figure 4 reports 3D-rendered imaging with 3D-Doctor (Able Software Corporation, Lexington, MA) on which light and dark grey colours represent the GDC and LSM phases, respectively.

TPB calculation

Matrix image calculation was undertaken for each 1-bit image. The goal of this exercise was to transform the image into a matrix

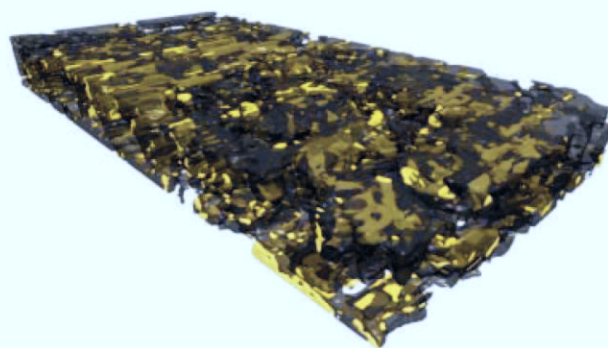


Figure 4. 3D structure rendering by 3D-Doctor software.

and then apply the developed algorithm for TPBL computation. It should be noted that in matrix image conversion, the (x,y) data were inverted to (y,x) , and this inversion was considered in the algorithm. MATLAB software from Mathworks served to build an algorithm for TPBL calculation from the image stack. The program, previously validated with a Ni-GDC virtual macroscopic model, was applied to SOFC anode simulation.

Sphere approximation

The algorithm produced the results in terms of voxel/total voxels. They did not show any significant dependence of magnification on voxel size estimation. Voxels should be converted to a more convenient unit such as nm^3/nm^3 . A sphere with the same outer surface as the voxel was used to convert the length of each voxel. This conversion method considers spheres with a similar outer surface. Thus, in our work, characteristic length was defined by the diameter of voxel surface-equivalent spheres.

RESULTS

Macroscopic Model

A 2-phase, solid-mixture model simulating Ni-GDC was constructed from wooden spheres encapsulated in acrylic resin. The algorithm was first applied to this Ni-GDC macroscopic model. The model's outer dimensions were $40 \text{ cm} \times 25 \text{ cm} \times 30 \text{ cm}$, from which 1.25-cm thick slices were removed, generating a total of 18 slices. Results obtained with the model were $\text{TPB} = 0.0025$ voxels/voxels³, demonstrating satisfactory algorithm operation.

FIB Cathode Sample

The measurement tool was then applied to a screen-printed LSM-GDC cathode substrate for TPBL assessment. Special attention was paid to the 'embedding' method to enhance the phase-differentiation capacity of the procedure. Moreover, the 'clarity' of the resultant images was optimised by contrast with cathode images. Figure 5 presents an image example from the stack obtained from the FIB with the TPBL measurement algorithm. Each captured image was about 993 pixels wide by 605 pixels high after alignment correction. A total of 18 images was acquired, and the TPBL result calculated for this structure was 0.006 voxels/voxels³. Resolution calculated from the images placed in the stack for the preliminary run was $922 \text{ nm}^3/\text{voxel}$. The estimated TPBL, with this characteristic length, was $86.5 \times 10^{-6} \text{ nm}/\text{nm}^3$.

Another set of tests was undertaken to improve repeatability of the previously obtained results. A screen-printed LSM-GDC

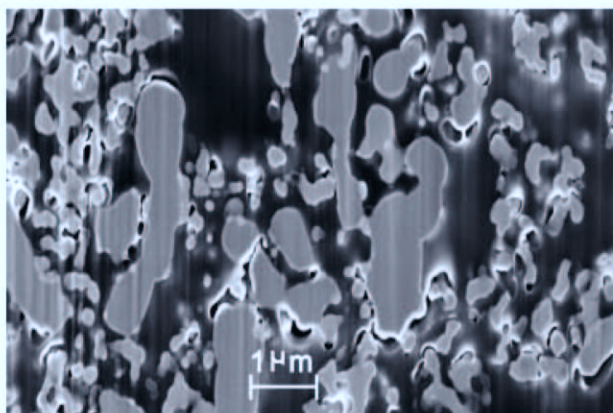


Figure 5. FIB image sample obtained by tomography.

cathode substrate made from nanopowders, a procedure described elsewhere (Wang et al., 1998; Xia et al., 2001), was studied in this run, with alignment prior to image capture to minimise translational drift. Each captured image was 748 pixels wide by 378 pixels high after alignment correction. Up to 80 images were taken for calculation, and the TPBL result was 0.014 voxels/total voxels, which corresponded to a value of 208 nm/nm³ for a total analysed volume of 1.275×10^9 nm³. This second attempt had enough tomographic pictures for the Hoshen–Kopelman algorithm. The calculated connected TPBL was 4.2% of total TPB sites in the sample.

Sensitivity Analysis

The algorithm was also applied to different image stacks, of which one to eight images were skipped in the *z* direction for TPBL calculation. The results revealed that the estimated TPBL decreased along with information reduction. To minimise correspondence problems, image spacing should be adjusted to about 1/4 of particle size (Bajaj et al., 1996). Sensitivity analysis disclosed that the calculated TPBL levelled off in the case of four skipped images, which corresponded to 40-nm spacing at the FIB. As illustrated in Figure 6, TPBL was underestimated when spacing between the images was higher than 40 nm. This was attributed to a considerable loss of precision when spacing approached the size of constitutive grains of the studied structure. In contrast, reducing image thickness between images should lead to a higher number of TPB sites, but minimal possible thickness depends on the apparatus. As spheres were used for the approximation of voxels, if the voxels were non-cubic, systematic error would depend on the spacing between images in the ‘*z*’ direction. Figure 7 extrapolates a ‘hypothetical cubic’ voxel to minimise systematic error. The extrapolated voxel volume ratio, that is the ‘number of occu-

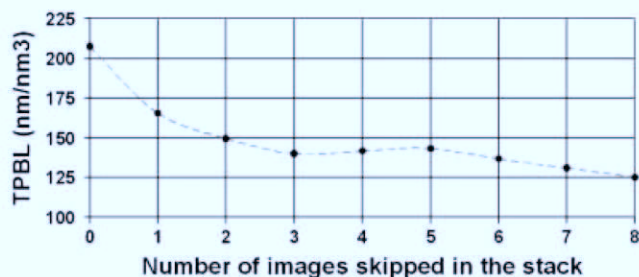


Figure 6. TPBL as a function of images skipped in the stack.

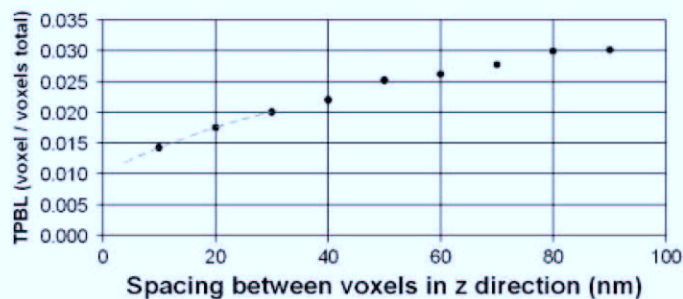


Figure 7. Spacing between voxels as a function of TPBL (voxels/total voxels).

ried voxels divided by the total number of voxels’ contained in the sample, was calculated to be 0.012 voxels/total voxels. TPBL, re-calculated on the basis of this ‘hypothetical cubic’ method, was 261 nm/nm³.

DISCUSSION

TPBL corresponds to the open interface between two different solid phases in multi-component materials. A tool was developed for the TPBL measurement of real, solid structures. Initially, the volume was sectioned by cropping each picture in the stack. It has been demonstrated that the number of TPB sites tends to zero. As expected physically, Figure 8 reveals that, as sectioning advances in the *x* or *y* direction, the method predicts a proportional reduction of TPB site numbers. Moreover, as expected, Figure 9 shows that, as the number of images is decreased in the *z* direction, the number of TPB sites tends to a minimum value when *z* = 1. A detailed zoom of previous figure is presented in Figure 10 that shows a minimum value of 543 pixels which corresponds to the TPB in the *z* = 1 plane between the GDC and LSM matrices. Also, as the reference surfaces are closely in contact between GDC and LSM, Figure 11 illustrates that the connectivity ultimately tends to 1, which means that all of the 543 TPB sites are connected electrically to both GDC and LSM reference surfaces. These results prove that the algorithm works well for real cathode structures.

The characterisation tool in this work is applicable to multi-phase materials that have components of different molecular weights. Porosity can also be measured with the same protocol and taking FIB apparatus limitations into account. Voxel resolution is dependent on FIB magnification and drift corrections.

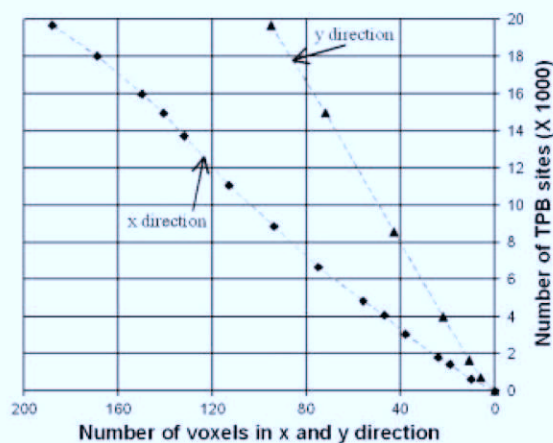


Figure 8. Number of TPB sites as a function of cropping matrix in *x* and *y* direction.

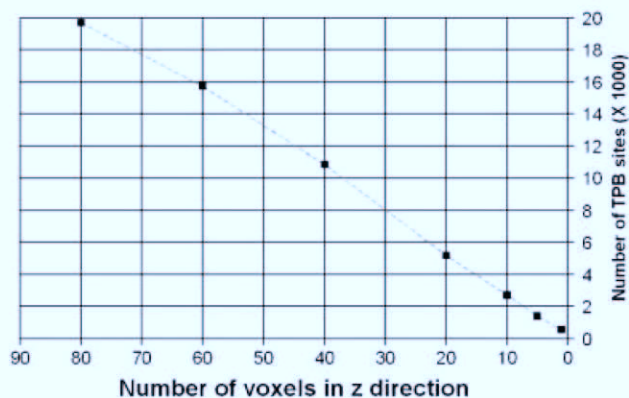


Figure 9. Number of TPB sites as a function of cropping matrix in z direction.

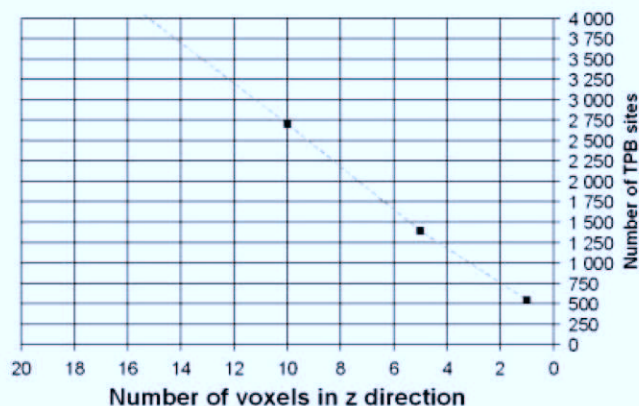


Figure 10. Detailed number of TPB sites as a function of cropping matrix in z direction.

Statistically, it is important to emphasise that high voxel resolution does not necessarily provide high accuracy, because the sampling volume is considerably reduced with increased magnification. Voxel resolution and sampling volume need to be balanced and adjusted to the form of the 3D structure studied. Although 'drift correction' of the FIB was applied, drift still represents a considerable source of error. The cross-linked LR white polymer resin and ceramic matrices augmented the drift problem owing to the 'charging effect' of electrostatic lens within the Crossbeam FIB apparatus. On the first attempt, translational alignment of

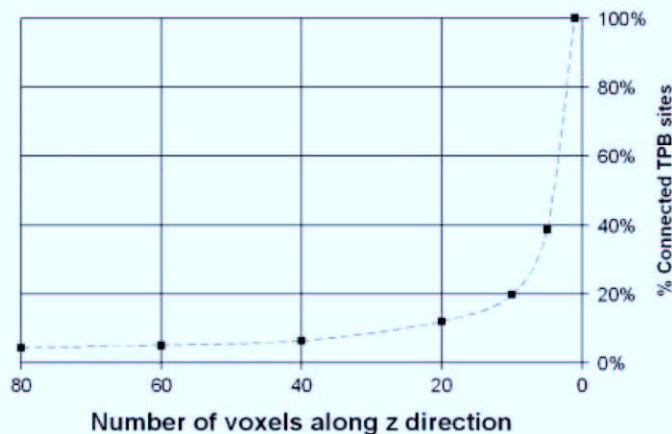


Figure 11. Connectivity of TPB sites as a function of cropping matrix in the z direction.

the FIB had to be repeated after five to six slices for drift correction; this drift diminished voxel resolution of the characterisation tool. Decreased voxel resolution was attributed to image alignment and 'cropping' for translational alignment. Cropping was a good way of keeping image information details intact during the translational correction process. The FIB image taken during this work was 1024×685 pixels in size and had to be 'cropped' to 993×605 pixels. An average loss of 14.4% of image information was, therefore, observed because of translational drift. A FIB equipped with an automatic drift correction device should be employed to maximise the voxel resolution of the characterisation tool. Another good solution for translational drift correction is the inclusion of alignment marks on each side of the etched volume of FIB sample. The second test of translational drift was reduced by etch alignment in the tomography procedure. The FIB image was 1024×685 pixels in size and had been cropped to 949×379 pixels for translation correction. In this case, an average 48.7% loss of image information was attributed to translation drift over the 80 tomographic pictures.

The first part of our work was undertaken to produce real TPBL calculation methodology. The developed prediction model relies on the ability to measure connected TPBL with tomography. In the case of a LSM-GDC cathode, used as a multi-component structure, it has been demonstrated that the compilation of at least 80 images is required for robust calculation. Our work improves the literature-available precision by a factor of almost 3.

The second part of our work defines the importance of systematic error in the actual approximation of voxels with spheres having the same outer diameter. Analysis of the second FIB-generated image stack shows that the 'hypothetical cubic-voxel' rather than the 'prismatic voxel' method significantly reduces the systematic error. Thus, re-calculated voxels change the initially calculated TPB by 11% at a TPBL of $231 \text{ nm}^3/\text{nm}^3$.

CONCLUSIONS

A tomographic approach was used for TPBL measurements in real 3D structures. The proposed protocol was validated in a Ni-GDC mock-up model, and then applied to a GDC-LSM cathode image stack by FIB nanotomography. LR white resin was very important in image separation prior to image capture. After initial image clarification and processing, an algorithm was built for TPBL measurement. Although sample charging and translational drifts are both considerable sources of error, resolution as high as $922 \text{ nm}^3/\text{voxel}$ was achieved with this technique on the first attempt. It was also shown that the TPBL data at the limits corresponded to the predicted results, proving the validity of the algorithm. Our findings could also be applied to any multi-phase material with different molecular weight components for porosity, tortuosity and TPBL calculation.

NOMENCLATURE

2D	two dimensions
3D	three dimensions
FIB	focused ion beam
GDC	gadolinium-doped ceria
LSM	lanthanum strontium manganate
MIP	mercury intrusion porosimetry
Ni	nickel
SEM	scanning electron microscopy
SOFC	solid oxide fuel cell
TPB	triple phase boundary

TPBL triple phase boundary length
YSZ yttrium stabilised zirconia
wt % Percent by weight

ACKNOWLEDGEMENTS

The authors are indebted to SOFC-Canada Network and the National Science and Engineering Research Council of Canada (NSERC) for funding related to this project. Also thanks to the technical contributions of Mr. Gilles Grondin, Mr. Jean Beerens and Mr. Henri Gauvin in impregnation, FIB services and glycine-nitrate process powder constitution, respectively. Special thanks are due to Mr. Ovid Da Silva for reviewing this manuscript.

REFERENCES

- Abbaspour, A., K. Nandakumar, J. Luo and K. T. Chuang, "A Novel Approach to Study the Structure Versus Performance Relationship of SOFC Electrodes," *J. Power Sources* **161**, 965–970 (2006).
- Bajaj, C. L., E. J. Coyle and K.-N. Lin, "Arbitrary Topology Shape Reconstruction from Planar Cross Sections," *Graph. Models Image Process.* **58**, 524–543 (1996).
- Bouchard, D., L. Sun, F. Gitzhofer and G. M. Brisard, Paper presented at the 207th Meeting of the Electrochemical Society, Quebec City, Quebec, Canada, May 15–20 (2005).
- Chavan, S. V. and A. K. Tyagi, "Preparation and Characterisation of $\text{Sr}_{0.09}\text{Ce}_{0.91}\text{O}_{1.91}$, SrCeO_3 , and Sr_2CeO_4 by Glycine-Nitrate Combustion: Crucial Role of Oxidant-to-Fuel Ratio," *J. Mater. Res.* **19**, 3181–3188 (2004).
- Gostovic, D., J. R. Smith, D. P. Kundinger, K. S. Jones and E. D. Wachsman, "Three-Dimensional Reconstruction of Porous LSCF Cathodes," *Electrochem. Solid-State Lett.* **10(B)**, 214–217 (2007).
- Holzer, L., F. Indutnyi, G. Gasser, B. Münch and M. Wegmann, "Three-Dimensional Analysis of Porous BaTiO_3 Ceramics Using FIB Nanotomography," *J. Microsc.* **216**, 84–95 (2004).
- Holzer, L., B. Münch, M. Wegmann, P. Gasser and R. J. Flatt, "FIB-Nanotomography of Particulate Systems Part I: Particle Shape and Topology of Interfaces," *J. Am. Ceram. Soc.* **89**, 2577–2585 (2006).
- Hoshen, J. and R. Kopelman, "Percolation and Cluster Distribution. I. Cluster Multiple Labeling Technique and Critical Concentration Algorithm," *Phys. Rev. B: Solid State* **14**, 3438–3445 (1976).
- Ji, Y., J. Liu, T. He, L. Cong, J. Wang and W. Su, "Single Intermedium-Temperature SOFC Prepared by Glycine-Nitrate Process," *J. Alloys Compd.* **353**, 257–262 (2003).
- Peng, R., C. Xia, Q. Fu, G. Meng and D. Peng, "Sintering and Electrical Properties of $(\text{CeO}_2)_{0.8}(\text{Sm}_2\text{O}_3)_{0.1}$ Powders Prepared by Glycine-Nitrate Process," *Mater. Lett.* **56**, 1043–1047 (2002a).
- Peng, R., C. Xia, X. Liu, D. Peng and G. Meng, "Intermediate-Temperature SOFCs with Thin $\text{Ce}_{0.8}\text{Y}_{0.2}\text{O}_{1.9}$ Films Prepared by Screen-Printing," *Solid State Ionics* **152–153**, 561–565 (2002b).
- Peng, R., C. Xia, D. Peng and G. Meng, "Effect of Powder Preparation on $(\text{CeO}_2)_{0.8}(\text{Sm}_2\text{O}_3)_{0.1}$ Thin Film Properties by Screen-Printing," *Mater. Lett.* **58**, 604–608 (2004).
- Schneider, L. C. R., C. L. Martin, Y. Bultel, D. Bouvard and E. Siebert, "Discrete Modelling of the Electrochemical Performance of SOFC Electrodes," *Electrochim. Acta* **52**, 314–324 (2006).

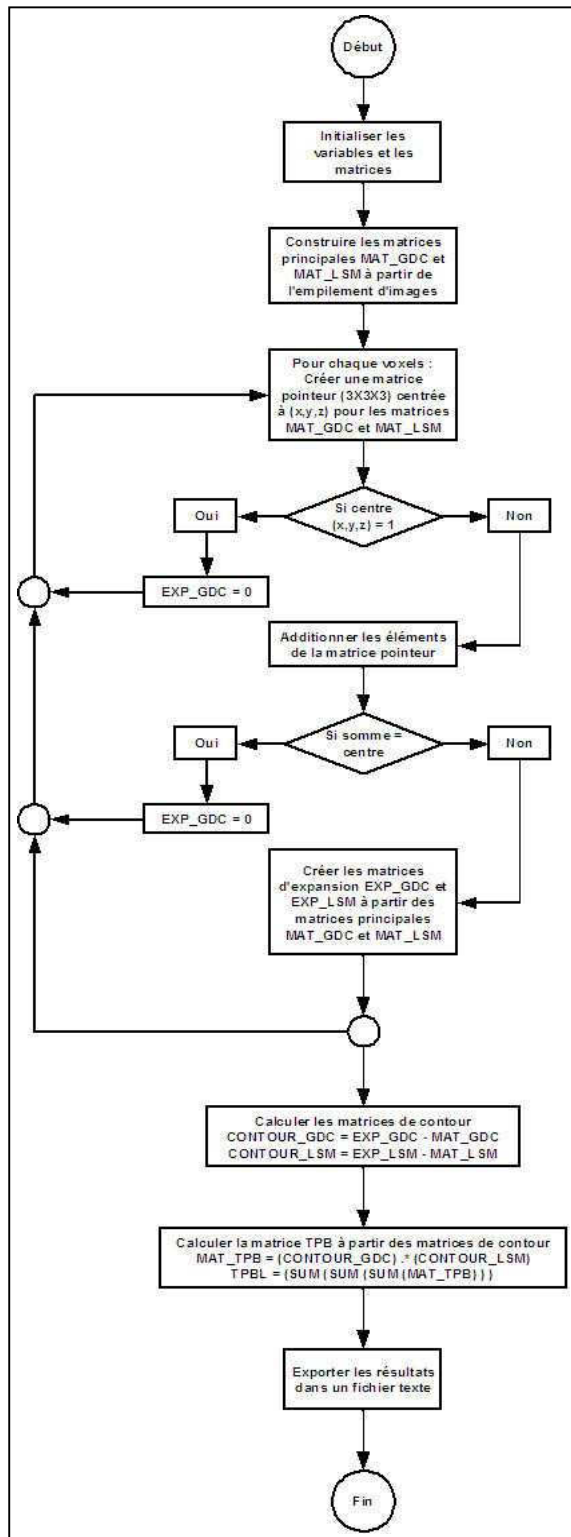
- Tsoga, A., A. Naoumidis, A. Gupta and D. Stoeber, "Microstructure and Interdiffusion Phenomena in YSZ-CGO Composite Electrolyte," *Mater. Sci. Forum* **308–311**, 794–799 (1999).
- Wang, S., Y. Jiang, Y. Zhang, J. Yan and W. Li, "Promoting Effect of YSZ on the Electrochemical Performance of YSZ + LSM Composite Electrodes," *Solid State Ionics* **113–115**, 291–303 (1998).
- Wilson, J. R., W. Kobsiriphat, R. Mendoza, H. Y. Chen, J. M. Hiller, D. J. Miller, K. Thornton, P. W. Voorhees, S. B. Adler and S. A. Barnett, "Three-Dimensional Reconstruction of a Solid-Oxide Fuel-Cell Anode," *Nat. Mater.* **5**, 541–544 (2006).
- Xia, C., F. Chen and M. Liu, "Reduced-Temperature Solid Oxide Fuel Cells Fabricated by Screen Printing," *Electrochem. Solid-State Lett.* **4**, 52–54 (2001).

Manuscript received December 8, 2010; revised manuscript received January 31, 2011; accepted for publication February 14, 2011.

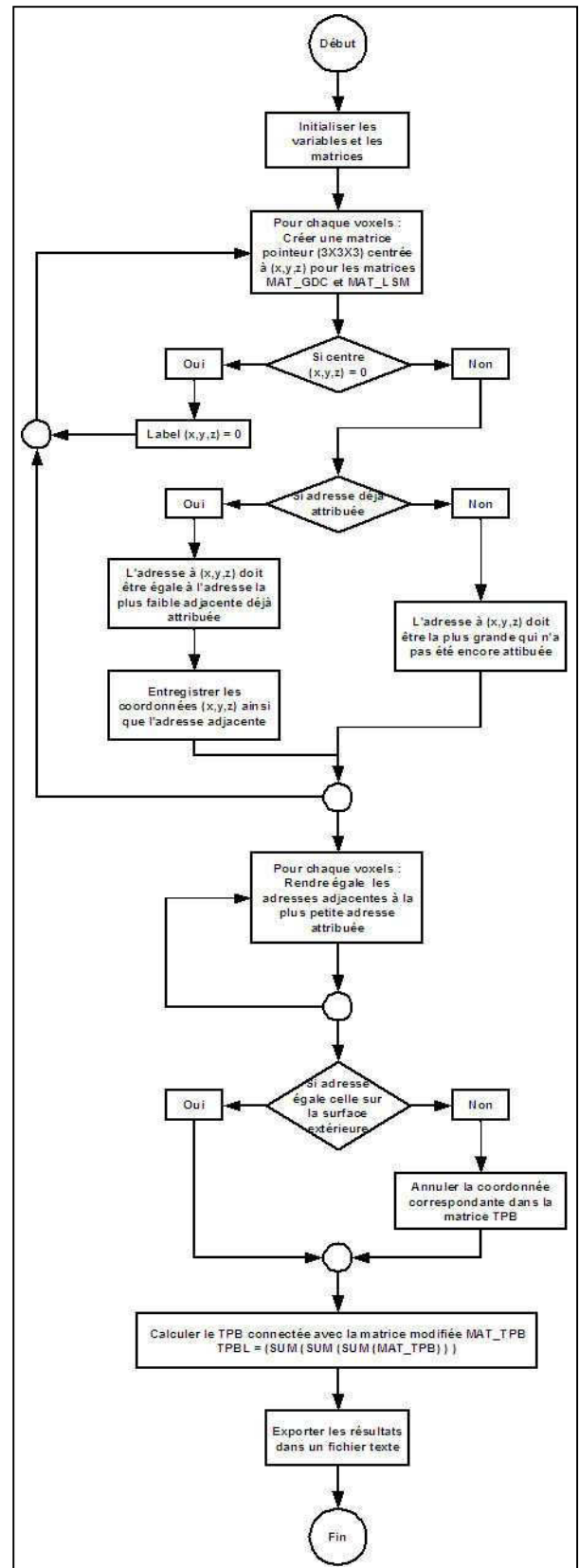
ANNEXE B

Graphe logique des principaux sous-programmes

Graphe logique de l'algorithme TPB



Graphe logique de l'algorithme Hoshen-Kopelman



ANNEXE C

Algorithme du programme Matlab

```

clc
clear all;
%//// Filter CASE 1 & Filter CASE 2 & Directionality\\\\\\%
xmax = double (input ('Quel est le nombre total de photos ? '));
ymax = double (input ('Quelle est la hauteur de chaque photos ? '));
zmax = double (input ('Quelle est la largeur de chaque photos ? '));

L = double (input ('Quelle distance entre chaque photos en microns
(selon x) ? '));
h = double (input ('Quelle est la hauteur de chaque voxels en microns
(selon y) ? '));
e = double (input ('Quelle est la largeur de chaque voxels en microns
(selon z) ? '));

iso_x = 0;
iso_y = 0;
iso_z = 0;
iso_xy = 0;
iso_xz = 0;
iso_yz = 0;

'i; tape 1 : Calcul de la matrice expansion du GDC'

0
mat_gdc_read = logical(int8(zeros(xmax,ymax,zmax)));
for ii = 1:xmax
    if ii < 10
        file_tmp = strcat('I000',int2str(ii));
        file = strcat(file_tmp, '.bmp');
    elseif ii >= 10 && ii < 100
        file_tmp = strcat('I00',int2str(ii));
        file = strcat(file_tmp, '.bmp');
    elseif ii >=100 && ii < 1000
        file_tmp = strcat('I0',int2str(ii));
        file = strcat(file_tmp, '.bmp');
    end
    mat_gdc_read(ii, :, :) = logical(int8(imread(file)));
end

exp_gdc = logical( zeros(xmax+2, ymax+2, zmax+2));

mat_gdc = logical(zeros(xmax+2,ymax+2,zmax+2));

for z = 1 : zmax
    for y = 1 : ymax
        for x = 1 : xmax
            mat_gdc (x+1, y+1, z+1) = mat_gdc_read (x,y,z);
        end
    end
end

imwrite(double(mat_gdc(:, :, 4)), 'test_mat_gdc0.bmp');
imwrite(double(mat_gdc(:, :, 9)), 'test_mat_gdc1.bmp');
imwrite(double(mat_gdc(:, :, 14)), 'test_mat_gdc2.bmp');
imwrite(double(mat_gdc(:, :, 19)), 'test_mat_gdc3.bmp');
imwrite(double(mat_gdc(:, :, 24)), 'test_mat_gdc4.bmp');
imwrite(double(mat_gdc(:, :, 29)), 'test_mat_gdc5.bmp');
imwrite(double(mat_gdc(:, :, 34)), 'test_mat_gdc6.bmp');
imwrite(double(mat_gdc(:, :, 39)), 'test_mat_gdc7.bmp');
imwrite(double(mat_gdc(:, :, 44)), 'test_mat_gdc8.bmp');

```

```

imwrite(double(mat_gdc(:,:,49)), 'test_mat_gdc9.bmp');

clear mat_gdc_read

mat_gdc(1, :, :) = 0;
mat_gdc(xmax+2, :, :) = 0;
mat_gdc(:, 1, :) = 0;
mat_gdc(:, ymax+2, :) = 0;
mat_gdc(:, :, 1) = 0;
mat_gdc(:, :, zmax+2) = 0;

for z = 2 : zmax + 1
    z_exp_gdc = z
    for y = 2 : ymax + 1
        for x = 2 : xmax + 1
            ref (1,1,1) = double(mat_gdc (x-1, y-1, z-1));
            ref (1,1,2) = double(mat_gdc (x-1, y-1, z));
            ref (1,1,3) = double(mat_gdc (x-1, y-1, z+1));
            ref (1,2,1) = double(mat_gdc (x-1, y, z-1));
            ref (1,2,2) = double(mat_gdc (x-1, y, z));
            ref (1,2,3) = double(mat_gdc (x-1, y, z+1));
            ref (1,3,1) = double(mat_gdc (x-1, y+1, z-1));
            ref (1,3,2) = double(mat_gdc (x-1, y+1, z));
            ref (1,3,3) = double(mat_gdc (x-1, y+1, z+1));
            ref (2,1,1) = double(mat_gdc (x, y-1, z-1));
            ref (2,1,2) = double(mat_gdc (x, y-1, z));
            ref (2,1,3) = double(mat_gdc (x, y-1, z+1));
            ref (2,2,1) = double(mat_gdc (x, y, z-1));
            ref (2,2,2) = double(mat_gdc (x, y, z));
            ref (2,2,3) = double(mat_gdc (x, y, z+1));
            ref (2,3,1) = double(mat_gdc (x, y+1, z-1));
            ref (2,3,2) = double(mat_gdc (x, y+1, z));
            ref (2,3,3) = double(mat_gdc (x, y+1, z+1));
            ref (3,1,1) = double(mat_gdc (x+1, y-1, z-1));
            ref (3,1,2) = double(mat_gdc (x+1, y-1, z));
            ref (3,1,3) = double(mat_gdc (x+1, y-1, z+1));
            ref (3,2,1) = double(mat_gdc (x+1, y, z-1));
            ref (3,2,2) = double(mat_gdc (x+1, y, z));
            ref (3,2,3) = double(mat_gdc (x+1, y, z+1));
            ref (3,3,1) = double(mat_gdc (x+1, y+1, z-1));
            ref (3,3,2) = double(mat_gdc (x+1, y+1, z));
            ref (3,3,3) = double(mat_gdc (x+1, y+1, z+1));

            centre = double(ref (2,2,2));

            if (centre == 1)
                exp_gdc (x, y, z) = 1;
            end
            if (centre == 0)
                somme_rel = (sum(sum(sum(double(ref)))))/27;
                if (somme_rel == centre)
                    exp_gdc (x, y, z) = 0;
                end
                if (somme_rel ~= centre)
                    if (ref (1,1,1) == 1)
                        if (exp_gdc (x, y, z) == 1)
                            exp_gdc (x, y, z) = 1;
                        else
                            exp_gdc (x, y, z) = 0;
                        end
                    end
                end
            end
        end
    end
end

```

```
if (ref (1,1,2) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (1,1,3) == 1)
    if (exp_gdc (x, y, z) == 1)
        exp_gdc (x, y, z) = 1;
    else
        exp_gdc (x, y, z) = 0;
    end
end
if (ref (1,2,1) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (1,2,2) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (1,2,3) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref(1,3,1) == 1)
    if (exp_gdc (x, y, z) == 1)
        exp_gdc (x, y, z) = 1;
    else
        exp_gdc (x, y, z) = 0;
    end
end
if (ref(1,3,2) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref(1,3,3) == 1)
    if (exp_gdc (x, y, z) == 1)
        exp_gdc (x, y, z) = 1;
    else
        exp_gdc (x, y, z) = 0;
    end
end
if (ref(2,1,1) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (2,1,2) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (2,1,3) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (2,2,1) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (2,2,3) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (2,3,1) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (2,3,2) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (2,3,3) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (3,1,1) == 1)
```

```

        if (exp_gdc (x, y, z) == 1)
            exp_gdc (x, y, z) = 1;
        else
            exp_gdc (x, y, z) = 0;
        end
    end
end
if (ref (3,1,2) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (3,1,3) == 1)
    if (exp_gdc (x, y, z) == 1)
        exp_gdc (x, y, z) = 1;
    else
        exp_gdc (x, y, z) = 0;
    end
end
if (ref (3,2,1) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (3,2,2) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (3,2,3) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (3,3,1) == 1)
    if (exp_gdc (x, y, z) == 1)
        exp_gdc (x, y, z) = 1;
    else
        exp_gdc (x, y, z) = 0;
    end
end
if (ref (3,3,2) == 1)
    exp_gdc (x, y, z) = 1;
end
if (ref (3,3,3) == 1)
    if (exp_gdc (x, y, z) == 1)
        exp_gdc (x, y, z) = 1;
    else
        exp_gdc (x, y, z) = 0;
    end
end
end
end
end
end
exp_gdc(1, :, :) = 0;
exp_gdc(xmax+2, :, :) = 0;
exp_gdc(:, 1, :) = 0;
exp_gdc(:, ymax+2, :) = 0;
exp_gdc(:, :, 1) = 0;
exp_gdc(:, :, zmax+2) = 0;

imwrite(double(exp_gdc(:, :, 4)), 'test_exp_gdc0.bmp');
imwrite(double(exp_gdc(:, :, 9)), 'test_exp_gdc1.bmp');
imwrite(double(exp_gdc(:, :, 14)), 'test_exp_gdc2.bmp');
imwrite(double(exp_gdc(:, :, 19)), 'test_exp_gdc3.bmp');
imwrite(double(exp_gdc(:, :, 24)), 'test_exp_gdc4.bmp');
imwrite(double(exp_gdc(:, :, 29)), 'test_exp_gdc5.bmp');
imwrite(double(exp_gdc(:, :, 34)), 'test_exp_gdc6.bmp');

```

```

imwrite(double(exp_gdc(:, :, 39)), 'test_exp_gdc7.bmp');
imwrite(double(exp_gdc(:, :, 44)), 'test_exp_gdc8.bmp');
imwrite(double(exp_gdc(:, :, 49)), 'test_exp_gdc9.bmp');

clear centre
clear ref

'i; tape 2 : Calcul de la matrice expansion du LSM'

mat_lsm_read = logical(int8(zeros(xmax, ymax, zmax)));
for ii = 1:xmax
    if ii < 10
        file_tmp = strcat('E000', int2str(ii));
        file = strcat(file_tmp, '.bmp');
    elseif ii >= 10 && ii < 100
        file_tmp = strcat('E00', int2str(ii));
        file = strcat(file_tmp, '.bmp');
    elseif ii >= 100 && ii < 1000
        file_tmp = strcat('E0', int2str(ii));
        file = strcat(file_tmp, '.bmp');
    end
    mat_lsm_read(ii, :, :) = logical(int8(imread(file)));
end

exp_lsm = logical( zeros(xmax+2, ymax+2, zmax+2));

mat_lsm = logical(zeros(xmax+2, ymax+2, zmax+2));

for z = 1 : zmax
    for y = 1 : ymax
        for x = 1 : xmax
            mat_lsm(x+1, y+1, z+1) = mat_lsm_read(x, y, z);
        end
    end
end

imwrite(double(mat_lsm(:, :, 4)), 'test_mat_lsm0.bmp');
imwrite(double(mat_lsm(:, :, 9)), 'test_mat_lsm1.bmp');
imwrite(double(mat_lsm(:, :, 14)), 'test_mat_lsm2.bmp');
imwrite(double(mat_lsm(:, :, 19)), 'test_mat_lsm3.bmp');
imwrite(double(mat_lsm(:, :, 24)), 'test_mat_lsm4.bmp');
imwrite(double(mat_lsm(:, :, 29)), 'test_mat_lsm5.bmp');
imwrite(double(mat_lsm(:, :, 34)), 'test_mat_lsm6.bmp');
imwrite(double(mat_lsm(:, :, 39)), 'test_mat_lsm7.bmp');
imwrite(double(mat_lsm(:, :, 44)), 'test_mat_lsm8.bmp');
imwrite(double(mat_lsm(:, :, 49)), 'test_mat_lsm9.bmp');

clear mat_lsm_read

mat_lsm(1, :, :) = 0;
mat_lsm(xmax+2, :, :) = 0;
mat_lsm(:, 1, :) = 0;
mat_lsm(:, ymax+2, :) = 0;
mat_lsm(:, :, 1) = 0;
mat_lsm(:, :, zmax+2) = 0;

for z = 2 : zmax + 1
    z_exp_lsm = z
    for y = 2 : ymax + 1
        for x = 2 : xmax + 1

```

```

ref (1,1,1) = double(mat_lsm (x-1, y-1, z-1));
ref (1,1,2) = double(mat_lsm (x-1, y-1, z));
ref (1,1,3) = double(mat_lsm (x-1, y-1, z+1));
ref (1,2,1) = double(mat_lsm (x-1, y, z-1));
ref (1,2,2) = double(mat_lsm (x-1, y, z));
ref (1,2,3) = double(mat_lsm (x-1, y, z+1));
ref (1,3,1) = double(mat_lsm (x-1, y+1, z-1));
ref (1,3,2) = double(mat_lsm (x-1, y+1, z));
ref (1,3,3) = double(mat_lsm (x-1, y+1, z+1));
ref (2,1,1) = double(mat_lsm (x, y-1, z-1));
ref (2,1,2) = double(mat_lsm (x, y-1, z));
ref (2,1,3) = double(mat_lsm (x, y-1, z+1));
ref (2,2,1) = double(mat_lsm (x, y, z-1));
ref (2,2,2) = double(mat_lsm (x, y, z));
ref (2,2,3) = double(mat_lsm (x, y, z+1));
ref (2,3,1) = double(mat_lsm (x, y+1, z-1));
ref (2,3,2) = double(mat_lsm (x, y+1, z));
ref (2,3,3) = double(mat_lsm (x, y+1, z+1));
ref (3,1,1) = double(mat_lsm (x+1, y-1, z-1));
ref (3,1,2) = double(mat_lsm (x+1, y-1, z));
ref (3,1,3) = double(mat_lsm (x+1, y-1, z+1));
ref (3,2,1) = double(mat_lsm (x+1, y, z-1));
ref (3,2,2) = double(mat_lsm (x+1, y, z));
ref (3,2,3) = double(mat_lsm (x+1, y, z+1));
ref (3,3,1) = double(mat_lsm (x+1, y+1, z-1));
ref (3,3,2) = double(mat_lsm (x+1, y+1, z));
ref (3,3,3) = double(mat_lsm (x+1, y+1, z+1));

centre = double(ref (2,2,2));

if (centre == 1)
    exp_lsm (x, y, z) = 1;
end
if (centre == 0)
    somme_rel = (sum(sum(sum(double(ref)))))/27;
    if (somme_rel == centre)
        exp_lsm (x, y, z) = 0;
    end
    if (somme_rel ~= centre)
        if (ref (1,1,1) == 1)
            if (exp_lsm (x, y, z) == 1)
                exp_lsm (x, y, z) = 1;
            else
                exp_lsm (x, y, z) = 0;
            end
        end
        if (ref (1,1,2) == 1)
            exp_lsm (x, y, z) = 1;
        end
        if (ref (1,1,3) == 1)
            if (exp_lsm (x, y, z) == 1)
                exp_lsm (x, y, z) = 1;
            else
                exp_lsm (x, y, z) = 0;
            end
        end
        if (ref (1,2,1) == 1)
            exp_lsm (x, y, z) = 1;
        end
        if (ref (1,2,2) == 1)
            exp_lsm (x, y, z) = 1;
        end
    end
end

```



```

end
if (ref (1,2,3) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref(1,3,1) == 1)
    if (exp_lsm (x, y, z) == 1)
        exp_lsm (x, y, z) = 1;
    else
        exp_lsm (x, y, z) = 0;
    end
end
if (ref(1,3,2) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref(1,3,3) == 1)
    if (exp_lsm (x, y, z) == 1)
        exp_lsm (x, y, z) = 1;
    else
        exp_lsm (x, y, z) = 0;
    end
end
if (ref(2,1,1) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (2,1,2) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (2,1,3) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (2,2,1) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (2,2,3) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (2,3,1) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (2,3,2) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (2,3,3) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (3,1,1) == 1)
    if (exp_lsm (x, y, z) == 1)
        exp_lsm (x, y, z) = 1;
    else
        exp_lsm (x, y, z) = 0;
    end
end
if (ref (3,1,2) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (3,1,3) == 1)
    if (exp_lsm (x, y, z) == 1)
        exp_lsm (x, y, z) = 1;
    else
        exp_lsm (x, y, z) = 0;
    end
end

```

```

end
if (ref (3,2,1) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (3,2,2) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (3,2,3) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (3,3,1) == 1)
    if (exp_lsm (x, y, z) == 1)
        exp_lsm (x, y, z) = 1;
    else
        exp_lsm (x, y, z) = 0;
    end
end
if (ref (3,3,2) == 1)
    exp_lsm (x, y, z) = 1;
end
if (ref (3,3,3) == 1)
    if (exp_lsm (x, y, z) == 1)
        exp_lsm (x, y, z) = 1;
    else
        exp_lsm (x, y, z) = 0;
    end
end
end
end
end
end
end

exp_lsm(1, :, :) = 0;
exp_lsm(xmax+2, :, :) = 0;
exp_lsm(:, 1, :) = 0;
exp_lsm(:, ymax+2, :) = 0;
exp_lsm(:, :, 1) = 0;
exp_lsm(:, :, zmax+2) = 0;

imwrite(double(exp_lsm(:, :, 4)), 'test_exp_lsm0.bmp');
imwrite(double(exp_lsm(:, :, 9)), 'test_exp_lsm1.bmp');
imwrite(double(exp_lsm(:, :, 14)), 'test_exp_lsm2.bmp');
imwrite(double(exp_lsm(:, :, 19)), 'test_exp_lsm3.bmp');
imwrite(double(exp_lsm(:, :, 24)), 'test_exp_lsm4.bmp');
imwrite(double(exp_lsm(:, :, 29)), 'test_exp_lsm5.bmp');
imwrite(double(exp_lsm(:, :, 34)), 'test_exp_lsm6.bmp');
imwrite(double(exp_lsm(:, :, 39)), 'test_exp_lsm7.bmp');
imwrite(double(exp_lsm(:, :, 44)), 'test_exp_lsm8.bmp');
imwrite(double(exp_lsm(:, :, 49)), 'test_exp_lsm9.bmp');

mat_pores = (exp_gdc | exp_lsm) & ~(mat_gdc | mat_lsm);

mat_pores(1, :, :) = 0; % ***** Change !!!!
mat_pores(xmax+2, :, :) = 0; % ***** Change !!!!
mat_pores(:, 1, :) = 0; % ***** Change !!!!
mat_pores(:, ymax+2, :) = 0; % ***** Change !!!!
mat_pores(:, :, 1) = 0; % ***** Change !!!!

```

```

mat_pores(:, :, zmax+2) = 0; % *****
Change !!!!

mat_tpb = mat_pores & (exp_gdc & exp_lsm);
mat_tpb(1, :, :) = 0; % *****
mat_tpb(xmax+2, :, :) = 0; % *****
mat_tpb(:, 1, :) = 0; % *****
mat_tpb(:, ymax+2, :) = 0; % *****
mat_tpb(:, :, 1) = 0; % *****
mat_tpb(:, :, zmax+2) = 0; % *****
Change !!!!

compteur = 1;
for z = 2 : zmax + 1 % **** CHANGE!
    z_adr_tpbldirty = z
    for y = 2 : ymax + 1 % **** CHANGE!
        for x = 2 : xmax + 1 % **** CHANGE!
            if mat_tpb(x, y, z) == 1
                adresse_tpbldirty (compteur,1) = compteur; % ****
CHANGE !
                adresse_tpbldirty (compteur,2) = x; % **** CHANGE !
                adresse_tpbldirty (compteur,3) = y; % **** CHANGE !
                adresse_tpbldirty (compteur,4) = z; % **** CHANGE !
                compteur = compteur + 1;
            end
        end
    end
end
tpbldirty = sum(sum(sum(mat_tpb)))
%
dlmwrite('tpbl_total_dirty.out', double(adresse_tpbldirty), ','); % ****
CHANGE !

'i;½tape de nettoyage de la matrice TPB'
pause
%-----
% ///REVISED Filter 1 and 2 combined into one filter. October 3, 2008\\
%*****
%// filter algorithm added by Craig Metcalfe to eliminate extraneous
%voxels.
% Filter 1: Eliminates double tpb voxels where two solid phases are
% adjacent and both are exposed to a pore directly above.
% Filter 2: tpb voxel must share a face with one solid phase and either
an
% edge on that face with the opposite solid phase or a different face
with
% the opposite solid phase.
%-----

for kk = 2:zmax + 1
    for jj = 2:ymax+1
        for ii = 2:xmax+1
            if mat_tpb(ii, jj, kk)
                %/// Removed filter 1 \\
                %/ set flags to determine if a tpb face is shared with
lsm
                %or gdc voxel
                Gxb = 0;    Gxf = 0;    Lxb = 0;    Lxf = 0;
            end
        end
    end
end

```

```

Gyb = 0;    Gyf = 0;    Lyb = 0;    Lyf = 0;
Gzb = 0;    Gzf = 0;    Lzb = 0;    Lzf = 0;
%/ shares xb face gdc
if mat_gdc(ii-1,jj,kk); Gxb = 1;    end
%/ shares xf face gdc
if mat_gdc(ii+1,jj,kk); Gxf = 1;    end
%/ shares yb face gdc
if mat_gdc(ii,jj-1,kk); Gyb = 1;    end
%/ shares yf face gdc
if mat_gdc(ii,jj+1,kk); Gyf = 1;    end
%/ shares zb face gdc
if mat_gdc(ii,jj,kk-1); Gzb = 1;    end
%/ shares zf face gdc
if mat_gdc(ii,jj,kk+1); Gzf = 1;    end
%/ shares xb face lsm
if mat_lsm(ii-1,jj,kk); Lxb = 1;    end
%/ shares xf face lsm
if mat_lsm(ii+1,jj,kk); Lxf = 1;    end
%/ shares yb face lsm
if mat_lsm(ii,jj-1,kk); Lyb = 1;    end
%/ shares yf face lsm
if mat_lsm(ii,jj+1,kk); Lyf = 1;    end
%/ shares zb face lsm
if mat_lsm(ii,jj,kk-1); Lzb = 1;    end
%/ shares zf face lsm
if mat_lsm(ii,jj,kk+1); Lzf = 1;    end
%/ set flags to determine if a tpb edge is shared with

```

lsm

```

%of gdc voxel
Gxy1 = 0;    Gxy2 = 0;    Gxy3 = 0;    Gxy4 = 0;
Gyz1 = 0;    Gyz2 = 0;    Gyz3 = 0;    Gyz4 = 0;
Gxz1 = 0;    Gxz2 = 0;    Gxz3 = 0;    Gxz4 = 0;
Lxy1 = 0;    Lxy2 = 0;    Lxy3 = 0;    Lxy4 = 0;
Lyz1 = 0;    Lyz2 = 0;    Lyz3 = 0;    Lyz4 = 0;
Lxz1 = 0;    Lxz2 = 0;    Lxz3 = 0;    Lxz4 = 0;
%/ shares xy1 edge with gdc
if mat_gdc(ii-1,jj-1,kk); Gxy1 = 1;    end
%/ shares xy2 edge with gdc
if mat_gdc(ii+1,jj-1,kk); Gxy2 = 1;    end
%/ shares xy3 edge with gdc
if mat_gdc(ii-1,jj+1,kk); Gxy3 = 1;    end
%/ shares xy4 edge with gdc
if mat_gdc(ii+1,jj+1,kk); Gxy4 = 1;    end
%/ shares yz1 edge with gdc
if mat_gdc(ii,jj-1,kk-1); Gyz1 = 1;    end
%/ shares yz2 edge with gdc
if mat_gdc(ii,jj+1,kk-1); Gyz2 = 1;    end
%/ shares yz3 edge with gdc
if mat_gdc(ii,jj-1,kk+1); Gyz3 = 1;    end
%/ shares yz4 edge with gdc
if mat_gdc(ii,jj+1,kk+1); Gyz4 = 1;    end
%/ shares xz1 edge with gdc
if mat_gdc(ii-1,jj,kk-1); Gxz1 = 1;    end
%/ shares xz2 edge with gdc
if mat_gdc(ii+1,jj,kk-1); Gxz2 = 1;    end
%/ shares xz3 edge with gdc
if mat_gdc(ii-1,jj,kk+1); Gxz3 = 1;    end
%/ shares xz4 edge with gdc
if mat_gdc(ii+1,jj,kk+1); Gxz4 = 1;    end
%/ shares xy1 edge with lsm
if mat_lsm(ii-1,jj-1,kk); Lxy1 = 1;    end

```

```

%/ shares xy2 edge with lsm
if mat_lsm(ii+1,jj-1,kk); Lxy2 = 1; end
%/ shares xy3 edge with lsm
if mat_lsm(ii-1,jj+1,kk); Lxy3 = 1; end
%/ shares xy4 edge with lsm
if mat_lsm(ii+1,jj+1,kk); Lxy4 = 1; end
%/ shares yz1 edge with lsm
if mat_lsm(ii,jj-1,kk-1); Lyz1 = 1; end
%/ shares yz2 edge with lsm
if mat_lsm(ii,jj+1,kk-1); Lyz2 = 1; end
%/ shares yz3 edge with lsm
if mat_lsm(ii,jj-1,kk+1); Lyz3 = 1; end
%/ shares yz4 edge with lsm
if mat_lsm(ii,jj+1,kk+1); Lyz4 = 1; end
%/ shares xz1 edge with lsm
if mat_lsm(ii-1,jj,kk-1); Lxz1 = 1; end
%/ shares xz2 edge with lsm
if mat_lsm(ii+1,jj,kk-1); Lxz2 = 1; end
%/ shares xz3 edge with lsm
if mat_lsm(ii-1,jj,kk+1); Lxz3 = 1; end
%/ shares xz4 edge with lsm
if mat_lsm(ii+1,jj,kk+1); Lxz4 = 1; end
%/ set flags to determine if faces and edges are shared
%with pores
Pxb = 0; Pxf = 0;
% NEW (line 626 through 859) %///!!!!\\%
Pyb = 0; Pyf = 0;
Pzb = 0; Pzf = 0;
Pxy1 = 0; Pxy2 = 0; Pxy3 = 0; Pxy4 = 0;
Pyz1 = 0; Pyz2 = 0; Pyz3 = 0; Pyz4 = 0;
Pxz1 = 0; Pxz2 = 0; Pxz3 = 0; Pxz4 = 0;
%/ shares xb face with pore
if mat_pores(ii-1,jj,kk); Pxb = 1; end
%/ shares xf face with pore
if mat_pores(ii+1,jj,kk); Pxf = 1; end
%/ shares yb face with pore
if mat_pores(ii,jj-1,kk); Pyb = 1; end
%/ shares yf face with pore
if mat_pores(ii,jj+1,kk); Pyf = 1; end
%/ shares zb face with pore
if mat_pores(ii,jj,kk-1); Pzb = 1; end
%/ shares zf face with pore
if mat_pores(ii,jj,kk+1); Pzf = 1; end
%/ shares xy1 edge with pore
if mat_pores(ii-1,jj-1,kk); Pxy1 = 1; end
%/ shares xy2 edge with pore
if mat_pores(ii+1,jj-1,kk); Pxy2 = 1; end
%/ shares xy3 edge with pore
if mat_pores(ii-1,jj+1,kk); Pxy3 = 1; end
%/ shares xy4 edge with pore
if mat_pores(ii+1,jj+1,kk); Pxy4 = 1; end
%/ shares yz1 edge with pore
if mat_pores(ii,jj-1,kk-1); Pyz1 = 1; end
%/ shares yz2 edge with pore
if mat_pores(ii,jj+1,kk-1); Pyz2 = 1; end
%/ shares yz3 edge with pore
if mat_pores(ii,jj-1,kk+1); Pyz3 = 1; end
%/ shares yz4 edge with pore
if mat_pores(ii,jj+1,kk+1); Pyz4 = 1; end
%/ shares xz1 edge with pore
if mat_pores(ii-1,jj,kk-1); Pxz1 = 1; end

```

```

    %/ shares xz2 edge with pore
    if mat_pores(ii+1,jj,kk-1); Pxz2 = 1;    end
    %/ shares xz3 edge with pore
    if mat_pores(ii-1,jj,kk+1); Pxz3 = 1;    end
    %/ shares xz4 edge with pore
    if mat_pores(ii+1,jj,kk+1); Pxz4 = 1;    end
    f1 = 0; f2 = 0; f3 = 0;
    f4 = 0; f5 = 0; f6 = 0;
    f7 = 0; f8 = 0; f9 = 0;
    f10 =0; f11 =0; f12 =0;
    if Gxb && (any([Lxz1 Lxz3 Lxy1 Lxy3]) || ...
              any([Lyb Lyf Lzb Lzf]))
        f1 = 1;
    end
    if Gxf && (any([Lxz2 Lxz4 Lxy2 Lxy4]) || ...
              any([Lyb Lyf Lzb Lzf]))
        f2 = 1;
    end
    if Gyb && (any([Lyz1 Lyz3 Lxy1 Lxy2]) || ...
              any([Lxb Lxf Lzb Lzf]))
        f3 = 1;
    end
    if Gyf && (any([Lyz2 Lyz4 Lxy3 Lxy4]) || ...
              any([Lxb Lxf Lzb Lzf]))
        f4 = 1;
    end
    if Gzb && (any([Lyz1 Lyz2 Lxz1 Lxz2]) || ...
              any([Lxb Lxf Lyb Lyf]))
        f5 = 1;
    end
    if Gzf && (any([Lyz3 Lyz4 Lxz3 Lxz4]) || ...
              any([Lxb Lxf Lyb Lyf Lzf]))
        f6 = 1;
    end
    %/ checklsm face shares. If no gdc edge share, delete
    if Lxb && (any([Gxz1 Gxz3 Gxy1 Gxy3]) || ...
              any([Gyb Gyf Gzb Gzf]))
        f7 = 1;
    end
    if Lxf && (any([Gxz2 Gxz4 Gxy2 Gxy4]) || ...
              any([Gyb Gyf Gzb Gzf]))
        f8 = 1;
    end
    if Lyb && (any([Gyz1 Gyz3 Gxy1 Gxy2]) || ...
              any([Gxb Gxf Gzb Gzf]))
        f9 = 1;
    end
    if Lyf && (any([Gyz2 Gyz4 Gxy3 Gxy4]) || ...
              any([Gxb Gxf Gzb Gzf]))
        f10 = 1;
    end
    if Lzb && (any([Gyz1 Gyz2 Gxz1 Gxz2]) || ...
              any([Gxb Gxf Gyb Gyf]))
        f11 = 1;
    end
    if Lzf && (any([Gyz3 Gyz4 Gxz3 Gxz4]) || ...
              any([Gxb Gxf Gyb Gyf]))
        f12 = 1;
    end
    if any([f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11 f12])

```

```

%-----
% share xf face with another tpb voxel && ...
% ((-z direction has two opposite phase lsm OR gdc
voxel
% && +z direction has two pore voxels && none of xb
yb
% or yf conditions above are satisfied) || ...
% ((+z direction has two opposite phase lsm OR gdc
voxel
% && -z direction has two pore voxels && none of xb
yb
% or yf conditions above are satisfied) || ...
% (-y direction has two opposite phase lsm OR gdc
% voxel && +y direction has two pore voxels && none
of
% xb zb or zf conditions above are satisfied) ||
...
% (+y direction has two opposite phase lsm OR gdc
% voxel && -y direction has two pore voxels && none
of
% xb zb or zf conditions above are satisfied)
if mat_tpb(ii+1,jj,kk) && ...
    (((Lzb && Gxz2) || (Gzb && Lxz2)) && (Pzf
&& Pxz4) && ~any([f1 f3 f4 f7 f9 f10])) || ...
    (((Lzf && Gxz4) || (Gzf && Lxz4)) && (Pzb
&& Pxz2) && ~any([f1 f3 f4 f7 f9 f10])) || ...
    (((Lyb && Gxy2) || (Gyb && Lxy2)) && (Pyf
&& Pxy4) && ~any([f1 f5 f6 f7 f11 f12])) || ...
    (((Lyf && Gxy4) || (Gyf && Lxy4)) && (Pyb
&& Pxy2) && ~any([f1 f5 f6 f7 f11 f12])))
    mat_tpb(ii,jj,kk) = 0;
    continue
end
% share xb face with another tpb voxel && ...
% ((-z direction has two opposite phase lsm OR gdc
voxel
% && +z direction has two pore voxels && none of xf
yb
% or yf conditions above are satisfied) || ...
% ((+z direction has two opposite phase lsm OR gdc
voxel
% && -z direction has two pore voxels && none of xf
yb
% or yf conditions above are satisfied) || ...
% (-y direction has two opposite phase lsm OR gdc
% voxel && +y direction has two pore voxels && none
of
% xf zb or zf conditions above are satisfied) ||
...
% (+y direction has two opposite phase lsm OR gdc
% voxel && -y direction has two pore voxels && none
of
% xf zb or zf conditions above are satisfied)
if mat_tpb(ii-1,jj,kk) && ...
    (((Lzb && Gxz1) || (Gzb && Lxz1)) && (Pzf
&& Pxz3) && ~any([f2 f3 f4 f8 f9 f10])) || ...
    (((Lzf && Gxz3) || (Gzf && Lxz3)) && (Pzb
&& Pxz1) && ~any([f2 f3 f4 f8 f9 f10])) || ...
    (((Lyb && Gxy1) || (Gyb && Lxy1)) && (Pyf
&& Pxy3) && ~any([f2 f5 f6 f8 f11 f12])) || ...

```

```

                                (((Lyf && Gxy3) || (Gyf && Lxy3)) && (Pyb
&& Pxy1) && ~any([f2 f5 f6 f8 f11 f12]))
                                mat_tpb(ii,jj,kk) = 0;
                                continue
                                end
                                % share yb face with another tpb voxel && ...
                                % ((-z direction has two opposite phase lsm OR gdc
voxel
                                % && +z direction has two pore voxels && none of yf
xb
                                % or xf conditions above are satisfied) || ...
                                % ((+z direction has two opposite phase lsm OR gdc
voxel
                                % && -z direction has two pore voxels && none of yf
xb
                                % or xf conditions above are satisfied) || ...
                                % (-x direction has two opposite phase lsm OR gdc
                                % voxel && +x direction has two pore voxels && none
of
                                % yf zb or zf conditions above are satisfied) ||
...
                                % (+x direction has two opposite phase lsm OR gdc
                                % voxel && -x direction has two pore voxels && none
of
                                % yf zb or zf conditions above are satisfied)
                                if mat_tpb(ii,jj-1,kk) && ...
                                    (((Lzb && Gyz1) || (Gzb && Lyz1)) && (Pzf
&& Pyz3) && ~any([f4 f1 f2 f10 f7 f8])) || ...
                                    (((Lzf && Gyz3) || (Gzf && Lyz3)) && (Pzb
&& Pyz1) && ~any([f4 f1 f2 f10 f7 f8])) || ...
                                    (((Lxb && Gxy1) || (Gxb && Lxy1)) && (Pxf
&& Pxy2) && ~any([f4 f5 f6 f10 f11 f12])) || ...
                                    (((Lxf && Gxy2) || (Gxf && Lxy2)) && (Pxb
&& Pxy1) && ~any([f4 f5 f6 f10 f11 f12]))
                                    mat_tpb(ii,jj,kk) = 0;
                                    continue
                                end
                                % share yf face with another tpb voxel && ...
                                % ((-z direction has two opposite phase lsm OR gdc
voxel
                                % && +z direction has two pore voxels && none of yb
xb
                                % or xf conditions above are satisfied) || ...
                                % ((+z direction has two opposite phase lsm OR gdc
voxel
                                % && -z direction has two pore voxels && none of yb
xb
                                % or xf conditions above are satisfied) || ...
                                % (-x direction has two opposite phase lsm OR gdc
                                % voxel && +x direction has two pore voxels && none
of
                                % yb zb or zf conditions above are satisfied) ||
...
                                % (+x direction has two opposite phase lsm OR gdc
                                % voxel && -x direction has two pore voxels && none
of
                                % yb zb or zf conditions above are satisfied)
                                if mat_tpb(ii,jj+1,kk) && ...
                                    (((Lzb && Gyz2) || (Gzb && Lyz2)) && (Pzf
&& Pyz4) && ~any([f3 f1 f2 f9 f7 f8])) || ...
                                    (((Lzf && Gyz4) || (Gzf && Lyz4)) && (Pzb

```



```

&& Pyz2) && ~any([f3 f1 f2 f9 f7 f8])) || ...
                (((Lxb && Gxy3) || (Gxb && Lxy3)) && (Pxf
&& Pxy4) && ~any([f3 f5 f6 f9 f11 f12])) || ...
                (((Lxf && Gxy4) || (Gxf && Lxy4)) && (Pxb
&& Pxy3) && ~any([f3 f5 f6 f9 f11 f12]))))
                mat_tpb(ii,jj,kk) = 0;
                continue
            end
            % share zb face with another tpb voxel && ...
            % ((-y direction has two opposite phase lsm OR gdc
voxel
            % && +y direction has two pore voxels && none of zf
xb
            % or xf conditions above are satisfied) || ...
            % ((+y direction has two opposite phase lsm OR gdc
voxel
            % && -y direction has two pore voxels && none of zf
xb
            % or xf conditions above are satisfied) || ...
            % (-x direction has two opposite phase lsm OR gdc
            % voxel && +x direction has two pore voxels && none
of
            % zf yb or yf conditions above are satisfied) ||
...
            % (+x direction has two opposite phase lsm OR gdc
            % voxel && -x direction has two pore voxels && none
of
            % zf yb or yf conditions above are satisfied))
            if mat_tpb(ii,jj,kk-1) && ...
                (((Lyb && Gyz1) || (Gyb && Lyz1)) && (Pyf
&& Pyz2) && ~any([f6 f1 f2 f12 f7 f8])) || ...
                (((Lyf && Gyz2) || (Gyf && Lyz2)) && (Pyb
&& Pyz1) && ~any([f6 f1 f2 f12 f7 f8])) || ...
                (((Lxb && Gxz1) || (Gxb && Lxz1)) && (Pxf
&& Pxz2) && ~any([f6 f3 f4 f12 f9 f10])) || ...
                (((Lxf && Gxz2) || (Gxf && Lxz2)) && (Pxb
&& Pxz1) && ~any([f6 f3 f4 f12 f9 f10]))))
                mat_tpb(ii,jj,kk) = 0;
                continue
            end
            % share zf face with another tpb voxel && ...
            % ((-y direction has two opposite phase lsm OR gdc
voxel
            % && +y direction has two pore voxels && none of zb
xb
            % or xf conditions above are satisfied) || ...
            % ((+y direction has two opposite phase lsm OR gdc
voxel
            % && -y direction has two pore voxels && none of zb
xb
            % or xf conditions above are satisfied) || ...
            % (-x direction has two opposite phase lsm OR gdc
            % voxel && +x direction has two pore voxels && none
of
            % zb yb or yf conditions above are satisfied) ||
...
            % (+x direction has two opposite phase lsm OR gdc
            % voxel && -x direction has two pore voxels && none
of
            % zb yb or yf conditions above are satisfied))
            if mat_tpb(ii,jj,kk-1) && ...

```



```

end
end
dlmwrite('porosity_total.out',double(adresse_porosity_total),',');

'i;½tape 5 : Calcul de la directionnaliti;½ de la matrice TPB'

clear mat_gdc
clear mat_lsm
length_cumul = 0;
compteur_cumul = 0;

for z = 2 : zmax + 1
    z_lgth_cumul = z
    for y = 2 : ymax + 1
        for x = 2 : xmax + 1
            compteur = 0;
            length = 0;
            % // set flags to identify the neighboring voxel positions
            % / set flags for face shares
            fxb = 0; fxf = 0;
            fyb = 0; fyf = 0;
            fzb = 0; fzf = 0;
            % / set flags for edge share parallel to z axis
            fxy1 = 0; fxy2 = 0;
            fxy3 = 0; fxy4 = 0;
            % / set flags for edge share parallel to y axis
            fxz1 = 0; fxz2 = 0;
            fxz3 = 0; fxz4 = 0;
            % / set flags for edge share parallel to x axis
            fyz1 = 0; fyz2 = 0;
            fyz3 = 0; fyz4 = 0;
            % / set flag for lone voxels
            fAlone = 0; length_A = 0;
            % // set counters for each face direction
            ll = zeros(1,6);
            % // set counters for each edge direction
            ee = zeros(1,12);
            f_overall = 0;
            %compteur = 0;
            %length = 0;
            if mat_tpb (x,y,z)
                % / xb
                if mat_tpb (x-1,y,z)
                    ll(1) = ll(1) + L/2;
                    fxb = 1;
                    iso_x = iso_x + 1;
                end
                % / xf
                if mat_tpb (x+1,y,z)
                    ll(2) = ll(2) + L/2;
                    fxf = 1;
                    iso_x = iso_x + 1;
                end
                % / yb
                if mat_tpb (x,y-1,z)
                    ll(3) = ll(3) + h/2;
                    fyb = 1;
                    iso_y = iso_y + 1;
                end
                % / yf
                if mat_tpb (x,y+1,z)

```

```

        ll(4) = ll(4) + h/2;
        fyf = 1;
        iso_y = iso_y + 1;
    end
    % / zb
    if mat_tpb (x,y,z-1)
        ll(5) = ll(5) + e/2;
        fzb = 1;
        iso_z = iso_z + 1;
    end
    % / zf
    if mat_tpb (x,y,z+1)
        ll(6) = ll(6) + e/2;
        fzf = 1;
        iso_z = iso_z + 1;
    end
    % / xy1
    if mat_tpb (x-1,y-1,z)
        ee(1) = ee(1) + sqrt(((L*L)/4)+((h*h)/4));
        fxy1 = 1;
        iso_xy = iso_xy + 1;
    end
    % / xy2
    if mat_tpb (x+1,y-1,z)
        ee(2) = ee(2) + sqrt(((L*L)/4)+((h*h)/4));
        fxy2 = 1;
        iso_xy = iso_xy + 1;
    end
    % / xy3
    if mat_tpb (x-1,y+1,z)
        ee(3) = ee(3) + sqrt(((L*L)/4)+((h*h)/4));
        fxy3 = 1;
        iso_xy = iso_xy + 1;
    end
    % / xy4
    if mat_tpb (x+1,y+1,z)
        ee(4) = ee(4) + sqrt(((L*L)/4)+((h*h)/4));
        fxy4 = 1;
        iso_xy = iso_xy + 1;
    end
    % / xz1
    if mat_tpb (x-1,y,z-1)
        ee(5) = ee(5) + sqrt(((L*L)/4)+((e*e)/4));
        fxz1 = 1;
        iso_xz = iso_xz + 1;
    end
    % / xz2
    if mat_tpb (x+1,y,z-1)
        ee(6) = ee(6) + sqrt(((L*L)/4)+((e*e)/4));
        fxz2 = 1;
        iso_xz = iso_xz + 1;
    end
    % / xz3
    if mat_tpb (x-1,y,z+1)
        ee(7) = ee(7) + sqrt(((L*L)/4)+((e*e)/4));
        fxz3 = 1;
        iso_xz = iso_xz + 1;
    end
    % / xz4
    if mat_tpb (x+1,y,z+1)
        ee(8) = ee(8) + sqrt(((L*L)/4)+((e*e)/4));
    end

```

```

        fxz4 = 1;
        iso_xz = iso_xz + 1;
    end
    % / yz1
    if mat_tpb (x,y-1,z-1)
        ee(9) = ee(9) + sqrt(((h*h)/4)+((e*e)/4));
        fyz1 = 1;
        iso_yz = iso_yz + 1;
    end
    % / yz2
    if mat_tpb (x,y+1,z-1)
        ee(10) = ee(10) + sqrt(((h*h)/4)+((e*e)/4));
        fyz2 = 1;
        iso_yz = iso_yz + 1;
    end
    % / yz3
    if mat_tpb (x,y-1,z+1)
        ee(11) = ee(11) + sqrt(((h*h)/4)+((e*e)/4));
        fyz3 = 1;
        iso_yz = iso_yz + 1;
    end
    % / yz4
    if mat_tpb (x,y+1,z+1)
        ee(12) = ee(12) + sqrt(((h*h)/4)+((e*e)/4));
        fyz4 = 1;
        iso_yz = iso_yz + 1;
    end
    end
    % / lone voxels
    if (mat_tpb (x-1,y,z) == 0) && ...
        (mat_tpb (x+1,y,z) == 0) && ...
        (mat_tpb (x,y-1,z) == 0) && ...
        (mat_tpb (x,y+1,z) == 0) && ...
        (mat_tpb (x,y,z-1) == 0) && ...
        (mat_tpb (x,y,z+1) == 0) && ...
        (mat_tpb (x-1,y-1,z) == 0) && ...
        (mat_tpb (x+1,y-1,z) == 0) && ...
        (mat_tpb (x-1,y+1,z) == 0) && ...
        (mat_tpb (x+1,y+1,z) == 0) && ...
        (mat_tpb (x-1,y,z-1) == 0) && ...
        (mat_tpb (x+1,y,z-1) == 0) && ...
        (mat_tpb (x-1,y,z+1) == 0) && ...
        (mat_tpb (x+1,y,z+1) == 0) && ...
        (mat_tpb (x,y-1,z-1) == 0) && ...
        (mat_tpb (x,y+1,z-1) == 0) && ...
        (mat_tpb (x,y-1,z+1) == 0) && ...
        (mat_tpb (x,y+1,z+1) == 0)
        length_A = sqrt((1/(2*pi))*((L*h) + (h*e) + (e*L)));
        fAlone = 1;
    end
    %/ shares only xb
    if fxb && (~fxf && ~fyb && ~fyf && ~fzb && ~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ll(1);
        continue
    end
    %/ shares only xf
    if fxf && (~fxb && ~fyb && ~fyf && ~fzb && ~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...

```

```

        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*11(2);
        continue
    end
    %/ shares only yb
    if fyb && (~fxf && ~fxb && ~fyf && ~fzb && ~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*11(3);
        continue
    end
    %/ shares only yf
    if yf && (~fxf && ~fxb && ~fyb && ~fzb && ~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*11(4);
        continue
    end
    %/ shares only zb
    if fzb && (~fxf && ~fxb && ~fyb && ~fyf && ~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*11(5);
        continue
    end
    %/ shares only zf
    if fzf && (~fxf && ~fxb && ~fyb && ~fyf && ~fzb) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*11(6);
        continue
    end
    %/ shares only xy1
    if fxy1 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(1);
        continue
    end
    %/ shares only xy2
    if fxy2 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy1 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(2);
        continue
    end
    %/ shares only xy3
    if fxy3 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy2 && ~fxy1 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(3);

```

```

        continue
    end
    %/ shares only xy4
    if fxy4 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy2 && ~fxy3 && ~fxy1 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(4);
        continue
    end
    %/ shares only xz1
    if fxz1 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(5);
        continue
    end
    %/ shares only xz2
    if fxz2 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(6);
        continue
    end
    %/ shares only xz3
    if fxz3 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz2 && ~fxz1 && ~fxz4)
        length_cumul = length_cumul + 2*ee(7);
        continue
    end
    %/ shares only xz4
    if fxz4 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz2 && ~fxz3 && ~fxz1)
        length_cumul = length_cumul + 2*ee(8);
        continue
    end
    %/ shares only yz1
    if fyz1 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(9);
        continue
    end
    %/ shares only yz2
    if fyz2 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz3 && ~fyz4 && ...

```

```

        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(10);
        continue
    end
    %/ shares only yz3
    if fyz3 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz2 && ~fyz1 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(11);
        continue
    end
    %/ shares only yz4
    if fyz4 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz2 && ~fyz3 && ~fyz1 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + 2*ee(12);
        continue
    end

    if ~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 ...
        && ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4 ...
        && ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4
        % / there are no edge shares
        % / check if only 2 face shares in x direction
        if (fxb && fxf) && (~fyb && ~fyf && ~fzb && ~fzf)
            length_cumul = length_cumul + sum([11(1) 11
(2)]);
                continue
            end
            % / check if only 2 face shares in y direction
            if (fyb && fyf) && (~fxb && ~fxf && ~fzb && ~fzf)
                length_cumul = length_cumul + sum([11(3) 11
(4)]);
                    continue
                end
                % / check if only 2 face shares in z direction
                if (fzb && fzf) && (~fxb && ~fxf && ~fyb && ~fyf)
                    length_cumul = length_cumul + sum([11(5) 11
(6)]);
                        continue
                    end
                    %/ Assumption is that there is never more than 2
faces
                    %shared for one tpb voxel
                end
                if ~fxb && ~fxf && ~fyb && ~fyf && ~fzb && ~fzf
                    %/ there are no face shares
                    %/ shares xy1 and xy4 only
                    if (fxy1 && fxy4) && (~fxy2 && ~fxy3 && ...
                        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
                        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
                        length_cumul = length_cumul + sum([ee(1) ee
(4)]);
                            continue
                        end
                        %/ shares xy2 and xy3 only
                        if (fxy2 && fxy3) && (~fxy1 && ~fxy4 && ...

```



```

        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + sum([ee(2) ee
(3)]);
        continue
    end
    %/ shares xz1 and xz4 only
    if (fxz1 && fxz4) && (~fxy1 && ~fxy2 && ~fxy3 &&
~fxy4 &&...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 &&...
        ~fxz2 && ~fxz3)
        length_cumul = length_cumul + sum([ee(5) ee
(8)]);
        continue
    end
    %/ shares xz2 and xz3 only
    if (fxz2 && fxz3) && (~fxy1 && ~fxy2 && ~fxy3 &&
~fxy4 &&...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 &&...
        ~fxz1 && ~fxz4)
        length_cumul = length_cumul + sum([ee(6) ee
(7)]);
        continue
    end
    %/ shares yz1 and yz4 only
    if (fyz1 && fyz4) && (~fxy1 && ~fxy2 && ~fxy3 &&
~fxy4 &&...
        ~fyz2 && ~fyz3 &&...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + sum([ee(9) ee
(12)]);
        continue
    end
    %/ shares yz2 and yz3 only
    if (fyz2 && fyz3) && (~fxy1 && ~fxy2 && ~fxy3 &&
~fxy4 &&...
        ~fyz1 && ~fyz4 &&...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul = length_cumul + sum([ee(10) ee
(11)]);
        continue
    end
    %/ Assumption is that no tpb voxel shares more than
2 edges
        %/ Check face and edge shares
    end
    if fAlone
        length_cumul = length_cumul + length_A;
        continue
    end
    %/ otherwise take average of all connections
    num = sum([ll ee]);
    denom = sum([fxb fxf fyb fyf fzb fzf fxy1 fxy2 fxy3 fxy4
...
        fxz1 fxz2 fxz3 fxz4 fyz1 fyz2 fyz3 fyz4]);
    if denom > 0
        length_cumul = length_cumul + num/denom;
    end
    compteur_cumul = compteur_cumul + compteur;
end
end
end

```

```

    end
end

length_cumul
pourcentage_isotropie_x = (iso_x / compteur_cumul) * 100
pourcentage_isotropie_y = (iso_y / compteur_cumul) * 100
pourcentage_isotropie_z = (iso_z / compteur_cumul) * 100
pourcentage_isotropie_xy = (iso_xy / compteur_cumul) * 100
pourcentage_isotropie_xz = (iso_xz / compteur_cumul) * 100
pourcentage_isotropie_yz = (iso_yz / compteur_cumul) * 100

compteur = 1;

increment = 1;
adresse_gdc = uint32(zeros(xmax+2,ymax+2,zmax+2));

'i; %tape 6 : Hoshen-Kopelman Matrice GDC'
for z = 1 : zmax + 1
    z_hk_gdc = z
    for y = 1 : ymax + 1
        for x = 1 : xmax + 1
            if exp_gdc(x,y,z) == 1
                adresse_gdc(x,y,z) = increment;
                if (exp_gdc(x-1,y,z) == 1 | exp_gdc(x,y-1,z) == 1 |
exp_gdc(x,y,z-1) == 1)
                    ordre = [ adresse_gdc(x-1,y,z); adresse_gdc(x,y-
1,z); adresse_gdc(x,y,z-1) ];
                    mat_ordre = sort (ordre);
                    if mat_ordre(3) > 0
                        if mat_ordre(2) > 0
                            if mat_ordre(1) > 0
                                adresse_gdc(x,y,z) = mat_ordre(1);
                            end
                            adresse_gdc(x,y,z) = mat_ordre(2);
                        end
                        adresse_gdc(x,y,z) = mat_ordre(3);
                    end
                    mat_ordre = [0; 0; 0];
                    ordre = [0; 0; 0];
                end
                if (exp_gdc(x-1,y,z) == 0 & exp_gdc(x,y-1,z) == 0 &
exp_gdc(x,y,z-1) == 0)
                    adresse_gdc(x,y,z) = increment;
                end
                increment = increment + 1;
            end
        end
    end
end

end

ordre = [0; 0; 0; 0];
mat_ordre = [0; 0; 0; 0];
compteur = 0;
repeater = 1;
adresse_gdc = uint32(zeros(xmax+2,ymax+2,zmax+2));
while repeater == 1
    repeater = 0;
    compteur = compteur + 1;
    cnvg_gdc = compteur
    for z = 2 : zmax + 1
        for y = 2 : ymax + 1

```

```

        for x = 2 : xmax + 1
            if adresse_gdc(x,y,z) > 1
                if (adresse_gdc(x-1,y,z) < adresse_gdc(x,y,z) &
adresse_gdc(x-1,y,z) > 0)
                    adresse_gdc(x,y,z) = adresse_gdc(x-1,y,z);
                    repeter = 1;
                end
                if (adresse_gdc(x,y-1,z) < adresse_gdc(x,y,z) &
adresse_gdc(x,y-1,z) > 0)
                    adresse_gdc(x,y,z) = adresse_gdc(x,y-1,z);
                    repeter = 1;
                end
                if (adresse_gdc(x,y,z-1) < adresse_gdc(x,y,z) &
adresse_gdc(x,y,z-1) > 0)
                    adresse_gdc(x,y,z) = adresse_gdc(x,y,z-1);
                    repeter = 1;
                end
                if (adresse_gdc(x+1,y,z) < adresse_gdc(x,y,z) &
adresse_gdc(x+1,y,z) > 0)
                    adresse_gdc(x,y,z) = adresse_gdc(x+1,y,z);
                    repeter = 1;
                end
                if (adresse_gdc(x,y+1,z) < adresse_gdc(x,y,z) &
adresse_gdc(x,y+1,z) > 0)
                    adresse_gdc(x,y,z) = adresse_gdc(x,y+1,z);
                    repeter = 1;
                end
                if (adresse_gdc(x,y,z+1) < adresse_gdc(x,y,z) &
adresse_gdc(x,y,z+1) > 0)
                    adresse_gdc(x,y,z) = adresse_gdc(x,y,z+1);
                    repeter = 1;
                end
            end
        end
    end
end
clear exp_gdc

increment = 1;
adresse_lsm = uint32(zeros(xmax+2,ymax+2,zmax+2));

'i; tape 7 : Hoshen-Kopelman Matrice LSM'

for z = 1 : zmax + 1
    z_hk_lsm = z
    for y = 1 : ymax + 1
        for x = 1 : xmax + 1
            if exp_lsm(x,y,z) == 1
                adresse_lsm(x,y,z) = increment;
                if (exp_lsm(x-1,y,z) == 1 | exp_lsm(x,y-1,z) == 1 |
exp_lsm(x,y,z-1) == 1)
                    ordre = [ adresse_lsm(x-1,y,z); adresse_lsm(x,y-
1,z); adresse_lsm(x,y,z-1) ];
                    mat_ordre = sort (ordre);
                    if mat_ordre(3) > 0
                        if mat_ordre(2) > 0
                            if mat_ordre(1) > 0
                                adresse_lsm(x,y,z) = mat_ordre(1);
                            end
                        end
                    end
                end
            end
        end
    end
end

```



```

        end
    end
end

clear exp_lsm
clear mat_tpb
mat_tpb = logical(uint32(zeros (xmax+2,ymax+2,zmax+2)));
compteur = 1;
for compteur = 1 : tpbl
    adr_tpbl = compteur
    xin = adresse_tpbl (compteur,2);
    yin = adresse_tpbl (compteur,3);
    zin = adresse_tpbl (compteur,4);
    for y = 3 : ymax
        for x = 3 : xmax
            if (adresse_lsm(x,y,2) == adresse_lsm(xin,yin,zin) &
adresse_gdc(x,y,zmax-1) == adresse_gdc(xin,yin,zin))
                mat_tpb(xin,yin,zin) = 1;
                continue
            end
        end
    end
end
clear adresse_gdc
clear adresse_lsm
adresse_pores = uint32(zeros (xmax+2,ymax+2,zmax+2));

'i;½tape 8 : Hoshen-Kopelman Matrice des pores'
increment = 1;
for z = 2 : zmax + 1
    z_hk_pores = z
    for y = 2 : ymax + 1
        for x = 2 : xmax + 1
            if mat_pores(x,y,z) == 1
                adresse_pores(x,y,z) = increment;
                if (mat_pores(x-1,y,z) == 1 | mat_pores(x,y-1,z) == 1 |
mat_pores(x,y,z-1) == 1)
                    ordre = [ adresse_pores(x-1,y,z); adresse_pores(x,y-
1,z); adresse_pores(x,y,z-1) ];
                    mat_ordre = sort (ordre);
                    if mat_ordre(3) > 0
                        if mat_ordre(2) > 0
                            if mat_ordre(1) > 0
                                adresse_pores(x,y,z) = mat_ordre(1);
                            end
                                adresse_pores(x,y,z) = mat_ordre(2);
                            end
                                adresse_pores(x,y,z) = mat_ordre(3);
                            end
                                mat_ordre = [0; 0; 0];
                                ordre = [0; 0; 0];
                            end
                                if (mat_pores(x-1,y,z) == 0 & mat_pores(x,y-1,z) == 0 &
mat_pores(x,y,z-1) == 0)
                                    adresse_pores(x,y,z) = increment;
                                end
                                    increment = increment + 1;
                                end
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
end
end

```

```

end

ordre = [0; 0; 0; 0];
mat_ordre = [0; 0; 0; 0];
compteur = 0;
repeter = 1;
while repeter == 1
    repeter = 0;
    compteur = compteur + 1;
    cnvg_pores = compteur
    for z = 2 : zmax + 1
        for y = 2 : ymax + 1
            for x = 2 : xmax + 1
                if adresse_pores(x,y,z) > 1
                    if (adresse_pores(x-1,y,z) < adresse_pores(x,y,z) &
adresse_pores(x-1,y,z) > 0)
                        adresse_pores(x,y,z) = adresse_pores(x-1,y,z);
                        repeter = 1;
                    end
                    if (adresse_pores(x,y-1,z) < adresse_pores(x,y,z) &
adresse_pores(x,y-1,z) > 0)
                        adresse_pores(x,y,z) = adresse_pores(x,y-1,z);
                        repeter = 1;
                    end
                    if (adresse_pores(x,y,z-1) < adresse_pores(x,y,z) &
adresse_pores(x,y,z-1) > 0)
                        adresse_pores(x,y,z) = adresse_pores(x,y,z-1);
                        repeter = 1;
                    end
                    if (adresse_pores(x+1,y,z) < adresse_pores(x,y,z) &
adresse_pores(x+1,y,z) > 0)
                        adresse_pores(x,y,z) = adresse_pores(x+1,y,z);
                        repeter = 1;
                    end
                    if (adresse_pores(x,y+1,z) < adresse_pores(x,y,z) &
adresse_pores(x,y+1,z) > 0)
                        adresse_pores(x,y,z) = adresse_pores(x,y+1,z);
                        repeter = 1;
                    end
                    if (adresse_pores(x,y,z+1) < adresse_pores(x,y,z) &
adresse_pores(x,y,z+1) > 0)
                        adresse_pores(x,y,z) = adresse_pores(x,y,z+1);
                        repeter = 1;
                    end
                end
            end
        end
    end
end
end
end
end
mat_pores = uint32(zeros(xmax+2,ymax+2,zmax+2));
for compteur = 1 : pores
    adr_por_total = compteur
    xin = adresse_porosity_total (compteur,2);
    yin = adresse_porosity_total (compteur,3);
    zin = adresse_porosity_total (compteur,4);
    for y = 3 : ymax
        for x = 3 : xmax
            if adresse_pores(x,y,2) == adresse_pores(xin,yin,zin)
                mat_pores(xin,yin,zin) = 1;
                continue
            end
        end
    end
end

```

```

        end
    end
end

'i; tape 9 : Exportation des r;ultats finaux'

clear adresse_pores
pores_ouverts = (sum(sum(sum(mat_pores))/(xmax*ymax*zmax)))*100 % -----
***** ----- ***** ----- This line changed !!!! % Craig added ) to
balance parenthesis 18/09/2008
pores_ouverts_sum = sum(sum(sum(mat_pores)));
mat_tpb_pores = uint32(double(mat_tpb) .* double(mat_pores));
compteur = 1;
adresse_open_porosity = zeros(pores_ouverts_sum,4);
for z = 2 : zmax + 1 % ***** CHANGE!
    adr_open_pore = z
    for y = 2 : ymax + 1 % ***** CHANGE!
        for x = 2 : xmax + 1 % ***** CHANGE!
            if mat_pores(x,y,z) == 1
                adresse_open_porosity (compteur,1) = compteur;
                adresse_open_porosity (compteur,2) = x;
                adresse_open_porosity (compteur,3) = y;
                adresse_open_porosity (compteur,4) = z;
                compteur = compteur + 1;
            end
        end
    end
end
adresse_open_porosity = uint32(adresse_open_porosity);
dlmwrite('porosity_open.out',double(adresse_open_porosity),' ');

activite = sum(sum(sum(mat_tpb_pores))) % changed to mat_tpb_pores
coefficient_activite = activite / tpbl
clear mat_tpb
clear mat_pores
compteur = 1;
adresse_tpb_act = zeros(activite,4);
for z = 2 : zmax + 1 % ***** CHANGE!
    adr_tpb_act = z
    for y = 2 : ymax + 1 % ***** CHANGE!
        for x = 2 : xmax + 1 % ***** CHANGE!
            if mat_tpb_pores(x,y,z) == 1
                adresse_tpb_act (compteur,1) = compteur;
                adresse_tpb_act (compteur,2) = x;
                adresse_tpb_act (compteur,3) = y;
                adresse_tpb_act (compteur,4) = z;
                compteur = compteur + 1;
            end
        end
    end
end
adresse_tpb_act = uint32(adresse_tpb_act);
dlmwrite('tpbl_active.out',double(adresse_tpb_act),' ');

length_cumul_final = 0;
compteur_cumul = 0;

for z = 2 : zmax + 1

```

```

lgth_cumul_act = z
for y = 2 : ymax + 1
    for x = 2 : xmax + 1
        compteur = 0;
        length = 0;
        % // set flags to identify the neighboring voxel positions
        % / set flags for face shares
        fxb = 0; fxf = 0;
        fyb = 0; fyf = 0;
        fzb = 0; fzf = 0;
        % / set flags for edge share parallel to z axis
        fxy1 = 0; fxy2 = 0;
        fxy3 = 0; fxy4 = 0;
        % / set flags for edge share parallel to y axis
        fxz1 = 0; fxz2 = 0;
        fxz3 = 0; fxz4 = 0;
        % / set flags for edge share parallel to x axis
        fyz1 = 0; fyz2 = 0;
        fyz3 = 0; fyz4 = 0;
        % / set flag for lone voxels
        fAlone = 0; length_A = 0;
        % // set counters for each face direction
        ll = zeros(1,6);
        % // set counters for each edge direction
        ee = zeros(1,12);
        f_overall = 0;
        %compteur = 0;
        %length = 0;
        if mat_tpb_pores (x,y,z)
            % / xb
            if mat_tpb_pores (x-1,y,z)
                ll(1) = ll(1) + L/2;
                fxb = 1;
                iso_x = iso_x + 1;
            end
            % / xf
            if mat_tpb_pores (x+1,y,z)
                ll(2) = ll(2) + L/2;
                fxf = 1;
                iso_x = iso_x + 1;
            end
            % / yb
            if mat_tpb_pores (x,y-1,z)
                ll(3) = ll(3) + h/2;
                fyb = 1;
                iso_y = iso_y + 1;
            end
            % / yf
            if mat_tpb_pores (x,y+1,z)
                ll(4) = ll(4) + h/2;
                fyf = 1;
                iso_y = iso_y + 1;
            end
            % / zb
            if mat_tpb_pores (x,y,z-1)
                ll(5) = ll(5) + e/2;
                fzb = 1;
                iso_z = iso_z + 1;
            end
            % / zf
            if mat_tpb_pores (x,y,z+1)

```



```

        ll(6) = ll(6) + e/2;
        fzf = 1;
        iso_z = iso_z + 1;
end
% / xy1
if mat_tpb_pores (x-1,y-1,z)
    ee(1) = ee(1) + sqrt(((L*L)/4)+((h*h)/4));
    fxy1 = 1;
    iso_xy = iso_xy + 1;
end
% / xy2
if mat_tpb_pores (x+1,y-1,z)
    ee(2) = ee(2) + sqrt(((L*L)/4)+((h*h)/4));
    fxy2 = 1;
    iso_xy = iso_xy + 1;
end
% / xy3
if mat_tpb_pores (x-1,y+1,z)
    ee(3) = ee(3) + sqrt(((L*L)/4)+((h*h)/4));
    fxy3 = 1;
    iso_xy = iso_xy + 1;
end
% / xy4
if mat_tpb_pores (x+1,y+1,z)
    ee(4) = ee(4) + sqrt(((L*L)/4)+((h*h)/4));
    fxy4 = 1;
    iso_xy = iso_xy + 1;
end
% / xz1
if mat_tpb_pores (x-1,y,z-1)
    ee(5) = ee(5) + sqrt(((L*L)/4)+((e*e)/4));
    fxz1 = 1;
    iso_xz = iso_xz + 1;
end
% / xz2
if mat_tpb_pores (x+1,y,z-1)
    ee(6) = ee(6) + sqrt(((L*L)/4)+((e*e)/4));
    fxz2 = 1;
    iso_xz = iso_xz + 1;
end
% / xz3
if mat_tpb_pores (x-1,y,z+1)
    ee(7) = ee(7) + sqrt(((L*L)/4)+((e*e)/4));
    fxz3 = 1;
    iso_xz = iso_xz + 1;
end
% / xz4
if mat_tpb_pores (x+1,y,z+1)
    ee(8) = ee(8) + sqrt(((L*L)/4)+((e*e)/4));
    fxz4 = 1;
    iso_xz = iso_xz + 1;
end
% / yz1
if mat_tpb_pores (x,y-1,z-1)
    ee(9) = ee(9) + sqrt(((h*h)/4)+((e*e)/4));
    fyz1 = 1;
    iso_yz = iso_yz + 1;
end
% / yz2
if mat_tpb_pores (x,y+1,z-1)
    ee(10) = ee(10) + sqrt(((h*h)/4)+((e*e)/4));

```

```

        fyz2 = 1;
        iso_yz = iso_yz + 1;
    end
    % / yz3
    if mat_tpb_pores (x,y-1,z+1)
        ee(11) = ee(11) + sqrt(((h*h)/4)+((e*e)/4));
        fyz3 = 1;
        iso_yz = iso_yz + 1;
    end
    % / yz4
    if mat_tpb_pores (x,y+1,z+1)
        ee(12) = ee(12) + sqrt(((h*h)/4)+((e*e)/4));
        fyz4 = 1;
        iso_yz = iso_yz + 1;
    end
    % / lone voxels
    if (mat_tpb_pores (x-1,y,z) == 0) && ...
        (mat_tpb_pores (x+1,y,z) == 0) && ...
        (mat_tpb_pores (x,y-1,z) == 0) && ...
        (mat_tpb_pores (x,y+1,z) == 0) && ...
        (mat_tpb_pores (x,y,z-1) == 0) && ...
        (mat_tpb_pores (x,y,z+1) == 0) && ...
        (mat_tpb_pores (x-1,y-1,z) == 0) && ...
        (mat_tpb_pores (x+1,y-1,z) == 0) && ...
        (mat_tpb_pores (x-1,y+1,z) == 0) && ...
        (mat_tpb_pores (x+1,y+1,z) == 0) && ...
        (mat_tpb_pores (x-1,y,z-1) == 0) && ...
        (mat_tpb_pores (x+1,y,z-1) == 0) && ...
        (mat_tpb_pores (x-1,y,z+1) == 0) && ...
        (mat_tpb_pores (x+1,y,z+1) == 0) && ...
        (mat_tpb_pores (x,y-1,z-1) == 0) && ...
        (mat_tpb_pores (x,y+1,z-1) == 0) && ...
        (mat_tpb_pores (x,y-1,z+1) == 0) && ...
        (mat_tpb_pores (x,y+1,z+1) == 0)
        length_A = sqrt((1/(2*pi))*((L*h) + (h*e) + (e*L)));
        fAlone = 1;
    end
    %/ shares only xb
    if fxb && (~fxf && ~fxb && ~fyf && ~fzb && ~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul_final = length_cumul_final + 2*11(1);
        continue
    end
    %/ shares only xf
    if fxf && (~fxb && ~fxb && ~fyf && ~fzb && ~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul_final = length_cumul_final + 2*11(2);
        continue
    end
    %/ shares only yb
    if fyb && (~fxf && ~fxb && ~fyf && ~fzb && ~fzf) && ...
        (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul_final = length_cumul_final + 2*11(3);
        continue
    end
end

```

```

%/ shares only yf
if fyf && (~fxf && ~fxb && ~fyb && ~fzb && ~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
     ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
     ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ll(4);
    continue
end
%/ shares only zb
if fzb && (~fxf && ~fxb && ~fyb && ~fyf && ~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
     ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
     ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ll(5);
    continue
end
%/ shares only zf
if fzf && (~fxf && ~fxb && ~fyb && ~fyf && ~fzb) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
     ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
     ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ll(6);
    continue
end
%/ shares only xy1
if fxy1 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy2 && ~fxy3 && ~fxy4 && ...
     ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
     ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(1);
    continue
end
%/ shares only xy2
if fxy2 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy1 && ~fxy3 && ~fxy4 && ...
     ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
     ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(2);
    continue
end
%/ shares only xy3
if fxy3 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy2 && ~fxy1 && ~fxy4 && ...
     ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
     ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(3);
    continue
end
%/ shares only xy4
if fxy4 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy2 && ~fxy3 && ~fxy1 && ...
     ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
     ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(4);
    continue
end
%/ shares only xz1

```

```

if fxz1 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
    ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
    ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(5);
    continue
end
%/ shares only xz2
if fxz2 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
    ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
    ~fxz1 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(6);
    continue
end
%/ shares only xz3
if fxz3 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
    ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
    ~fxz2 && ~fxz1 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(7);
    continue
end
%/ shares only xz4
if fxz4 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
    ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
    ~fxz2 && ~fxz3 && ~fxz1)
    length_cumul_final = length_cumul_final + 2*ee(8);
    continue
end
%/ shares only yz1
if fyz1 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
    ~fyz2 && ~fyz3 && ~fyz4 && ...
    ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(9);
    continue
end
%/ shares only yz2
if fyz2 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
    ~fyz1 && ~fyz3 && ~fyz4 && ...
    ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(10);
    continue
end
%/ shares only yz3
if fyz3 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
    ~fyz2 && ~fyz1 && ~fyz4 && ...
    ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(11);
    continue

```

```

end
%/ shares only yz4
if fyz4 && (~fxb && ~fxf && ~fyb && ~fyf && ~fzb &&
~fzf) && ...
    (~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 && ...
    ~fyz2 && ~fyz3 && ~fyz1 && ...
    ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + 2*ee(12);
    continue
end

if ~fxy1 && ~fxy2 && ~fxy3 && ~fxy4 ...
    && ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4 ...
    && ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4
    % / there are no edge shares
    % / check if only 2 face shares in x direction
    if (fxb && fxf) && (~fyb && ~fyf && ~fzb && ~fzf)
        length_cumul_final = length_cumul_final + sum
        ([l1(1) l1(2)]);
        continue
    end
    % / check if only 2 face shares in y direction
    if (fyb && fyf) && (~fxb && ~fxf && ~fzb && ~fzf)
        length_cumul_final = length_cumul_final + sum
        ([l1(3) l1(4)]);
        continue
    end
    % / check if only 2 face shares in z direction
    if (fzb && fzf) && (~fxb && ~fxf && ~fyb && ~fyf)
        length_cumul_final = length_cumul_final + sum
        ([l1(5) l1(6)]);
        continue
    end
    end
    %/ Assumption is that there is never more than 2
faces
    %shared for one tpb voxel
end
if ~fxb && ~fxf && ~fyb && ~fyf && ~fzb && ~fzf
    %/ there are no face shares
    %/ shares xy1 and xy4 only
    if (fxy1 && fxy4) && (~fxy2 && ~fxy3 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul_final = length_cumul_final + sum
        ([ee(1) ee(4)]);
        continue
    end
    %/ shares xy2 and xy3 only
    if (fxy2 && fxy3) && (~fxy1 && ~fxy4 && ...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 && ...
        ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
        length_cumul_final = length_cumul_final + sum
        ([ee(2) ee(3)]);
        continue
    end
    %/ shares xz1 and xz4 only
    if (fxz1 && fxz4) && (~fxy1 && ~fxy2 && ~fxy3 &&
~fxy4 &&...
        ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 &&...
        ~fxz2 && ~fxz3)
        length_cumul_final = length_cumul_final + sum

```

```

([ee(5) ee(8)]);
    continue
end
%/ shares xz2 and xz3 only
if (fxz2 && fxz3) && (~fxy1 && ~fxy2 && ~fxy3 &&
~fxy4 &&...
    ~fyz1 && ~fyz2 && ~fyz3 && ~fyz4 &&...
    ~fxz1 && ~fxz4)
    length_cumul_final = length_cumul_final + sum
([ee(6) ee(7)]);
    continue
end
%/ shares yz1 and yz4 only
if (fyz1 && fyz4) && (~fxy1 && ~fxy2 && ~fxy3 &&
~fxy4 &&...
    ~fyz2 && ~fyz3 &&...
    ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + sum
([ee(9) ee(12)]);
    continue
end
%/ shares yz2 and yz3 only
if (fyz2 && fyz3) && (~fxy1 && ~fxy2 && ~fxy3 &&
~fxy4 &&...
    ~fyz1 && ~fyz4 &&...
    ~fxz1 && ~fxz2 && ~fxz3 && ~fxz4)
    length_cumul_final = length_cumul_final + sum
([ee(10) ee(11)]);
    continue
end
%/ Assumption is that no tpb voxel shares more than
2 edges
    %/ Check face and edge shares
end
if fAlone
    length_cumul_final = length_cumul_final + length_A;
    continue
end
%/ otherwise take average of all connections
num = sum([ll ee]);
denom = sum([fxb fxf fyb fyf fzb fzf fxy1 fxy2 fxy3 fxy4
...
    fxz1 fxz2 fxz3 fxz4 fyz1 fyz2 fyz3 fyz4]);
if denom > 0
    length_cumul_final = length_cumul_final + num/denom;
end
compteur_cumul = compteur_cumul + compteur;
end
end
end
end

%end    Commented out: Craig 18/09/2008
length_cumul_final

'fin du calcul'
save workspace4.mat

```