Développement de la scatterométrie dynamique pour le suivi en temps réel de procédés. Application à la microélectronique.

Sébastien Soulan

THÈSE

présentée par

Sébastien SOULAN

pour obtenir le titre de
DOCTEUR de l'UNIVERSITÉ GRENOBLE I - JOSEPH FOURIER

École doctorale Électronique, Électrotechnique, Automatique & Traitement du Signal
Spécialité Micro et Nano Électronique

Développement de la scatterométrie dynamique pour le suivi en temps réel de procédés. Application à la microélectronique.

Thèse dirigée par M. Patrick SCHIAVONE et encadrée par M. Maxime BESACIER
Date de soutenance : 8 décembre 2008

Composition du jury:

M. Pierre JOUVELOT	**Rapporteur**
M. Antonello DE MARTINO	**Rapporteur**
M. Gérard GRANET	**Examinateur**
M. Jérôme HAZART	**Examinateur**

Invité : M. Vincent FARYS

RÉSUMÉ

La métrologie *in situ* et le contrôle de procédés en temps réel sont pour l'industrie de la microélectronique des enjeux d'une importance cruciale. Une technique de caractérisation optique basée sur une analyse de la lumière diffractée par un objet, la scatterométrie, fait preuve pour cela d'un potentiel remarquable. Il s'agit d'une méthode non destructive qui permet de mesurer indirectement et avec excellente précision des grandeurs géométriques de motifs périodiques.

Pour la résolution de ce problème inverse, il est coutume de comparer une signature relevée par ellipsométrie (par exemple) avec une bibliothèque de signatures optiques calculées au préalable. Dans cette thèse, ce principe appliqué couramment en situation statique (mesure en ligne d'un échantillon) a été étendu à une application dynamique (suivi de procédés en temps réel), pour laquelle les signatures sont acquises avec une faible résolution en longueurs d'onde mais avec une grande fréquence. Ces développements ont consisté d'une part en l'élaboration d'un algorithme de reconstruction de forme basé sur la régularisation de Tikhonov et d'autre part sur l'utilisation d'une architecture de calcul particulière, les processeurs graphiques (GPU).

A des fins de mise au point et de validation, nous nous sommes appuyés sur des procédés de la microélectronique pour lesquels le suivi en temps réel est un défi majeur pour le futur : gravure de résine par plasma et fluage de résine pour la nano-impression.

Mots clés : scatterométrie, problème inverse dynamique, temps réel, reconstruction de forme, processeurs graphiques, microélectronique.

SUMMARY

Development of dynamic scatterometry for real time process control.
Applications for microelectronics.

In situ metrology and real time process control are fundamental challenges for the future of the microelectronics industry. For this purpose, scatterometry, which is an optical characterisation technique based on scattered light seems to have a great potential. It is a non destructive method that allows indirect measurements of periodic patterns. For solving this inverse problem, one use to compare an acquired optical signature (from an ellipsometer, for example) with a large set of pre-computed signatures : the library. In this thesis, this principle, which is commonly applied in static conditions (on line sample measurements), is expanded to dynamic applications (real time process control) for which acquired signatures have a low resolution but a high acquisition frequency. These developments are based, on the one hand, on a new shape reconstruction algorithm (inspired by the Tikhonov regularization) and on the other hand, on a innovative computing architecture : the graphics processing units (GPU). For adjustment and validation purpose, we have dealt with microelectronics processes for which real time monitoring is key for the future : plasma etch process (namely resist trimming) and resist reflow for nano-imprint.

Keywords : scatterometry, dynamic inverse problem, real time, shape reconstruction, graphics processors, microelectronics.

Thèse effectuée au Laboratoire des Technologies de la Microélectronique de Grenoble.

LTM - CNRS CEA/LETI/D2NT 17, av. des martyrs 38054 Grenoble Cedex

à Tatie,
à Papi...

Remerciements

En prélude à ce manuscrit, je voudrais remercier le plus chaleureusement possible Messieurs les membres du jury et leur témoigner mon admiration. Ils ont accepté d'évaluer mes travaux avec rigueur et compétence, et ce pourtant, en un temps remarquablement court.

Maintenant, au lecteur fasciné par ce manuscrit (si toutefois, il en naquit un jour) qui voudrait savoir ce que fut la vie du thésard qui en est l'auteur, je parlerai d'une période enchantée, assurément la plus enrichissante qui soit pour l'étudiant que j'étais...

Je lui parlerai d'un encadrement idéal : d'un tuteur Maxime et d'un directeur de thèse Patrick toujours aimables, disponibles, compétents et curieux de tout ; et n'ayant pour le thésard que je fus que de la confiance et de la bienveillance (Naïf comme je suis, je prends pour de la bienveillance le gavage impitoyable en remarques et corrections subi lors de la rédaction de cette thèse et lors de la préparation à la soutenance...).

Je lui parlerai aussi d'un labo, le LTM. Exigu certes, mais peuplé de gens toujours sympathiques et disponibles, promptes à ripailler gaiment à une table commune du réfectoire.

Je mentionnerai ces collègues qui m'ont aidé à faire avancer mon travail en substituant notamment ma maladresse expérimentale à leur grande connaissance des outils de manips : je pense à Tanguy, Mohamed, Jean-Hervé... ou alors en m'ouvrant les yeux vers des domaines fascinants qui m'étaient inconnus auparavant : les statistiques de Jérôme, la combinatoire d'Aysé...

De quoi le rendre envieux, ce lecteur ! Et ce n'est pas fini : je lui parlerai des bons souvenirs en compagnie des topains (et topains assimilés) du bureau 217 : Malou, Max (again), Sylvaine, Coco, etc. Il est peut-être petit notre bureau mais l'on s'y sent bien ; il est chaleureux, on y rigole bien et tout le monde y est bienvenu.

Souvenirs inoubliables aussi de la compagnie de ces grands amateurs de cafés infects et thés insipides que sont notamment Florian et Jean-Raoul, de ces grandes discussions à défaire le monde (pour mieux le reconstruire un jour).

Et parce que trois ans de thèse ne se vivent pas qu'au labo, je lui expliquerai combien a compté ma famille : en étant unie dans les moments les plus difficiles, j'y ai trouvé le réconfort nécessaire pour finir ces travaux. Je remercierai Maman, Papa pour m'avoir fait ainsi (puissé-je convenir un petit peu au modèle de vertu et de courage qu'ils ont imaginé... Hum.) ainsi que frangine Virginie pour sa bonne humeur permanente pendant les vacances et son pot belge trop méconnu mais ô combien efficace : Westmalle Triple - Daskalidès.

Enfin, je lui parlerai d'Estelle qui a accepté de quitter la capitale pour rester à mes cotés. Puisse-t'elle ne jamais le regretter...

Table des matières

Introduction

L'invention du transistor, du circuit intégré, puis le développement de la microélectronique conjointement à celui de l'informatique qui s'en est suivi, a notoirement bouleversé nos sociétés économiquement développées. En 50 ans à peine, l'humanité a développé des outils de communication abolissant totalement les distances qui freinaient jadis les flux d'informations entre individus sur la planète.

L'utilisation journalière du réseau Internet ou des téléphones mobiles par exemple, est permise aujourd'hui grâce à des investissements massifs en recherche et développement dans le domaine de la microélectronique. Depuis le début, ces recherches visent à fabriquer des microprocesseurs moins chers et toujours plus performants en améliorant notamment :

- **L'architecture des circuits.** Le premier processeur, l'Intel 4004 (1971), disposant alors de 2300 transistors, savait exécuter 46 instructions sur 4 bits (chaque instruction occupait alors jusqu'à 12 cycles d'horloge). On peut le comparer à son homologue de 2008 (processeur pour ordinateurs personnels) : 500 millions de transistors, unités de calculs multiples reliées entre elles par des canaux ultra-rapides, plusieurs niveaux de mémoire cache, registres 64 bits, instructions optimisées en cycles d'horloge (éventuellement anticipées), unités de calcul vectoriel, etc.
- **La vitesse de fonctionnement.** La plupart des microprocesseurs ont un fonctionnement cadencé par une horloge. Elle fonctionnait à 740 KHz pour le 4004, à environ 100 MHz il y a une dizaine d'année, et aujourd'hui la norme est de 3 GHz.
- **La consommation énergétique.** Alors qu'un circuit plus gros et fonctionnant plus vite devrait consommer d'avantage, les progrès dans ce domaine font aujourd'hui qu'un processeur de téléphone mobile consomme quelques mW pour une puissance de calcul comparable à celle d'un ordinateur personnel des années 90.

Toutes ces améliorations ont permis de multiplier par un million la puissance des circuits intégrés en moins de quarante ans. Elles sont pour une très large part dues à la diminution de la dimension de la grille de l'élément de base : le **transistor**[1] (cf. fig. 1). Là fut et demeure toujours le point crucial du développement de la microélectronique.

La fabrication de ces composants élémentaires suppose non seulement des procédés très sophistiqués (lithographie, dépôts et gravure de matériaux, etc) mais aussi une grande maîtrise de la métrologie. À ce titre, la dimension critique du transistor (ou CD, *critical dimension*), définie

[1] On ne saurait ici ne pas mentionner aussi les progrès faits en science des matériaux, qui ont participé tout autant à la réduction des dimensions qu'à l'amélioration de la consommation énergétique et des vitesses de fonctionnement.

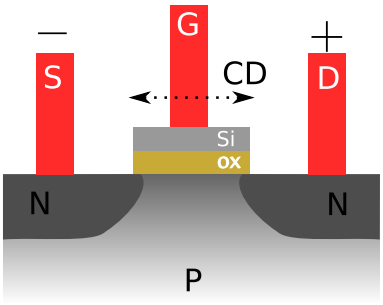


Fig. 1: Transistor. Il s'agit de l'équivalent d'un interrupteur : C'est une tension appliquée à la grille qui commande le passage du courant entre la source est le drain. Cet élément est constitutif de toutes les fonctions logiques d'un circuit semi-conducteur, et la diminution de sa largeur de grille (CD) notamment, est cruciale.

communément par sa largeur de grille fait l'objet des plus grandes attentions : l'amélioration des performances recherchée implique en particulier une capacité à mesurer cette largeur, aussi petite soit elle, avec une grande précision.

Les principaux acteurs du secteur, associés dans l'ITRS (*International Technology Roadmap on Semiconductors*), publient chaque année un ensemble de recommandations (pour l'année 2007, voir [9]) destiné à guider le développement technologique du secteur ; ils définissent notamment tous les 3 ans des **nœuds** technologiques qui sont les buts à atteindre pour la R&D. Ces nœuds, symbolisés par une grandeur en nanomètres, correspondent à la demi-période d'une mémoire DRAM (*Dynamic Random Access Memory*, il s'agit de la mémoire vive utilisée par exemple dans les ordinateurs).

Le tableau 1 représente quantitativement les objectifs en matière de dimensions à fabriquer et à mesurer pour les nœuds technologiques en cours de production et à venir.

Année de production	2007	2010	2013	2016	2019
demi-période DRAM (nm)	65	45	32	22	16
grille de transistor, circuit logique (nm)	25	18	13	9	6
précision de la métrologie (3σ, nm)					
ligne isolée	0.52	0.37	0.27	**0.19**	**0.12**
lignes denses	1.26	0.84	0.58	0.42	**0.33**

Tab. 1: Extrait de la feuille de route ITRS 2007 - métrologie. Les recommandations de métrologie en caractères gras correspondent aux cas où il n'existe pas de solutions industrielles. Dans les autres cas, les solutions sont connues.

Pour arriver à ces objectifs, l'association recommande le développement de certains outils de métrologie. En annexe A nous avons représenté un extrait de la feuille de route 2007 : on remarque, parmi les techniques jugées les plus prometteuses et dont l'effort de mise au point en contexte industriel est fortement recommandé, **la scatterométrie**.

La scatterométrie est une méthode de métrologie qui s'appuie sur le phénomène de diffraction d'un rayon de lumière pour mesurer indirectement des grandeurs géométriques de très petites dimensions sur des motifs **périodiques**. Là est d'ailleurs sa principale exigence : la technique suppose aujourd'hui que l'on réserve sur la surface de la puce une zone test recouverte d'un réseau périodique. La mesure sur un véritable circuit en cours de fabrication (comme les mémoires, qui sont une juxtaposition

périodique d'unités de stockage de bits) fait partie des développements à venir.

La scatterométrie comporte de nombreux avantages pour la mesure d'un profil nanométrique en regard notamment des autres techniques utilisées aujourd'hui couramment par l'industrie : CDSEM (*CD Scanning Electron Microscopy*, microscopie électronique à balayage) ou AFM (*Atomic Force Microscopy*, microscopie à force atomique) :

- La scatterométrie impose seulement la présence d'un rayon lumineux ; elle est donc particulièrement adaptée aux mesures *in situ*, c'est-à-dire à l'endroit même où se produit l'une des étapes de fabrication. Le CDSEM ou l'AFM supposent au contraire que l'échantillon soit inséré dans leur propre environnement de mesure, et ne permettent donc au mieux que de la métrologie *en ligne*, c'est à dire intégrée à la chaîne de fabrication.
- La scatterométrie ne nécessite pas d'instrumentation très lourde. Un ellipsomètre ou un réflectomètre suffit pour effectuer plusieurs mesures ellipsométriques par seconde. C'est la raison pour laquelle la scatterométrie est vue comme une technique potentielle pour le suivi en **temps réel** d'un procédé.

La scatterométrie est donc candidate pour être la future technique de métrologie permettant un suivi de procédé *in situ* et en temps réel. Ces travaux se situent en amont d'une quelconque application industrielle ; ils sont prospectifs dans un domaine encore largement inexploité : le problème inverse en **scatterométrie dynamique**. Dans ce domaine, la complexité n'est plus dans l'instrumentation, mais dans le traitement rapide et efficace des données issues de celle-ci, l'ambition étant d'atteindre la plus grande précision de mesure possible pour satisfaire les recommandations des fabriquants.

Les travaux présentés ici apparaîtront au lecteur de diverses natures : physique, informatique, mathématiques appliquées. La combinaison des trois disciplines est apparue nécessaire devant la variété des défis que représentaient cette technique de métrologie et son évolution de statique en dynamique.

Ce rapport est découpé en quatre parties, chacune présentant différentes briques nécessaires au développement de la scatterométrie dynamique :

1. La première partie concerne la physique de la scatterométrie. Quels sont les phénomènes en jeu ? Que représentent les grandeurs mesurées ? Comment les mesure t-on ? Comment les simule t-on (problème direct) ?
2. La deuxième partie est liée au problème inverse, c'est-à-dire à l'utilisation de grandeurs mesurées pour déterminer les caractéristiques géométriques nanométriques de l'objet diffractant. Comment le résout-on habituellement ? Que proposer pour une résolution temps réel en situation dynamique ? Nous exposerons dans cette partie un nouvel algorithme de reconstruction dynamique du profil. Son efficacité sera notamment assurée par l'utilisation d'une architecture de calcul particulière : les processeurs graphiques.
3. La troisième partie sera dédiée à la validation expérimentale. Nous utiliserons pour cela une expérience fictive puis des expérimentations réelles comme un suivi de fluage de résine et un procédé élémentaire de gravure plasma (gravure de résine).
4. Une dernière partie sera dédiée à la détermination des indices optiques des matériaux. Ces grandeurs qui déterminent le comportement des matériaux utilisés lors d'une excitation électromagnétique représentent une donnée cruciale du problème direct. Leur détermination est un problème inverse et nous proposons de le résoudre en utilisant ce qui a été développé pour la scatterométrie dynamique.

Les travaux de cette thèse ont été permis par une collaboration entre un industriel du secteur, la société franco-italienne STMicroelectronics, et un acteur académique, le CNRS. Les développements présentés dans ce mémoire ont été effectués au LTM, le Laboratoire des Technologies de la Microélectronique, une structure sous tutelle essentiellement académique : CNRS, Université Joseph Fourier de Grenoble, Institut National Polytechnique de Grenoble.

Première partie

Ellipsométrie de diffraction de la lumière

Cette première partie traite des fondamentaux du domaine qui nous concerne. Nous partirons des phénomènes physiques les plus élementaires, le champ électromagnétique et ses variations, pour décrire ensuite les phénomènes d'onde lumineuse, de polarisation et de diffraction. Après cet aspect physique des choses, nous rendrons compte des moyens expérimentaux utilisés pour les mesures de grandeurs, notamment de polarisation. Pour finir cette partie, nous expliciterons les méthodes de calcul numérique permettant de simuler *directement* les phénomènes en question.

CHAPITRE 1

Rayonnement lumineux et interactions

Ce chapitre concerne la description des phénomènes physiques considérés dans cette thèse. L'exposé débutera par les aspects les plus fondamentaux comme le champ électromagnétique et sa propagation en onde lumineuse, puis nous décrirons les phénomènes de polarisation, de réfraction et de diffraction de la lumière.

1.1 La nature de la lumière

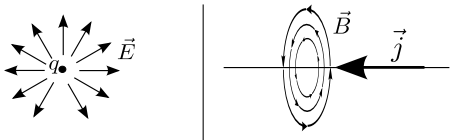
1.1.1 Les équations de Maxwell dans les milieux continus

En avril 1820, le danois Hans Christian Œrsted s'aperçut, lors d'un cours qu'il donnait sur l'électricité, que le passage d'un courant électrique dans un fil était capable de faire bouger l'axe d'une boussole. Il venait de découvrir l'interaction entre les phénomènes électriques et magnétiques.

Ses travaux furent largement diffusés et précédèrent tout ce qui, durant le XIX^{e}siècle, allait fonder l'électromagnétisme :

- En septembre 1820, le français André-Marie Ampère prend la suite des travaux du danois et énonce la règle selon laquelle le sens du courant qui circule dans le fil détermine l'orientation de l'axe de la boussole.
- En 1821, le physicien anglais Michael Faraday fait la démonstration de ce que qu'il appelle la rotation électromagnétique. Il s'agit en fait du moteur électrique, c'est-à-dire la rotation d'éléments magnétiques autour d'un fil électrique.
- En 1831, une collaboration entre les physiciens allemands Carl Friedrich Gauss et Wilhelm Weber aboutit à deux lois. L'une régit l'électrostatique en énonçant qu'une charge électrostatique produit un champ électrique divergent. L'autre loi, homologue pour le magnétisme, exprime qu'il n'existe pas de monopôle (de "charge") magnétique susceptible de créer une divergence de champ magnétique (La source la plus simple de champ magnétique, l'aimant, est nécessairement un dipôle constitué des pôles nord et sud).

En 1873, dans l'ouvrage *Electricity and Magnetism*, le physicien écossais James Clerk Maxwell unifie toutes ces lois et peu après, en 1884, Oliver Heaviside les formalise en quatre équations aux dérivées partielles couplées. Il s'agit des **équations de Maxwell**. Elles forment le postulat de l'électromagnétisme et, à elles seules, régissent localement l'interaction des champs électrique et magnétique :



(a) Une charge électrostatique localisée génère une divergence du champ électrique

(b) Un courant électrique génère un rotationnel de champ magnétique

Fig. 1.1: Représentations des champs couplés électriques et magnétiques

- Une charge électrostatique crée un champ électrique **divergent** (cf. fig. 1.1(a)). C'est l'équation de Maxwell-Gauss.
- Le mouvement d'un champ magnétique crée un rotationnel de champ électrique : équation de Maxwell-Faraday.
- Une source magnétique n'est pas unipolaire ; elle ne crée donc pas de divergence : équation de Maxwell-Thomson, nommée ainsi en l'honneur du physicien anglais Joseph John Thomson, découvreur du rayon cathodique (application du principe selon lequel une particule chargée voit sa trajectoire déviée par la présence d'un champ magnétique) et surtout de l'électron (prix Nobel 1906).
- Un courant électrique, provoqué par un déplacement de charge ou un mouvement du champ électrique, crée un rotationnel de champ magnétique (cf. fig. 1.1(b)). C'est l'équation de Maxwell-Ampère.

Ces équations, présentées ci-dessous, sont valables dans tous les milieux continus, c'est-à-dire des milieux présentant, à l'échelle macroscopique, des variations continues des grandeurs physiques (indices optiques par exemple). Cette approximation est valable tant que l'on ne s'approche pas des dimensions atomiques : la nature discontinue de la matière à cette échelle (atomes, électrons, et surtout vide) nécessiterait un modèle différent.

$$\begin{cases} \nabla \cdot \mathbf{D} = \rho & \text{Maxwell-Gauss} \quad (1.1) \\ \nabla \times \mathbf{E} = -\dfrac{\partial \mathbf{B}}{\partial t} & \text{Maxwell-Faraday} \quad (1.2) \\ \nabla \cdot \mathbf{H} = 0 & \text{Maxwell-Thomson} \quad (1.3) \\ \nabla \times \mathbf{H} = \mathbf{J} + \dfrac{\partial \mathbf{D}}{\partial t} & \text{Maxwell-Ampère} \quad (1.4) \end{cases}$$

Les grandeurs locales utilisées sont les suivantes :

- **D** est le vecteur **induction électrique**, il s'exprime en Coulomb par mètre carré ($C.m^{-2}$).
- **E** est le vecteur **champ électrique**, il s'exprime en Volt par mètre ($V.m^{-1}$).
- **B** est le vecteur **induction magnétique**, il s'exprime en Weber par mètre carré ($Wb.m^{-2}$) ou Tesla (T).
- **H** est le vecteur **champ magnétique**, il s'exprime en Ampère par mètre ($A.m^{-1}$).
- ρ est la **densité de charge électrique**, il s'exprime en Coulomb par mètre cube ($C.m^{-3}$).
- **J** est le vecteur **densité de courant**, il s'exprime en Ampère par mètre carré ($A.m^{-2}$).

1.1.1.1 Relations de passages

La continuité du milieu, hypothèse première des quatre équations précédentes, se brise lors du passage d'un milieu à un autre. Des relations permettent néanmoins de résoudre les équations de Maxwell le cas échéant ; elles donnent la valeur de la discontinuité des composantes normales ($\mathbf{n}_{12}\cdot$) et tangentielles ($\mathbf{n}_{12}\times$) induite par le changement de milieu.

En considérant la situation et les notations du schéma 1.2, ces relations sont, **à l'interface des milieux ① et ②** :

$$\begin{cases} \mathbf{n}_{12} \times (\mathbf{E}_2 - \mathbf{E}_1) = 0 \\ \mathbf{n}_{12} \times (\mathbf{H}_2 - \mathbf{H}_1) = \mathbf{J}_s \\ \mathbf{n}_{12} \cdot (\mathbf{D}_2 - \mathbf{D}_1) = \rho_s \\ \mathbf{n}_{12} \cdot (\mathbf{B}_2 - \mathbf{B}_1) = 0 \end{cases}$$

où, en plus des grandeurs déjà connues, on a noté :

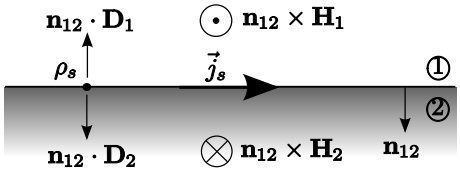


Fig. 1.2: Relations de passage à une interface : le champ magnétique est modifié par un courant surfacique et le champ électrique par une charge surfacique

- $\mathbf{J}_s$ le vecteur **densité de courant surfacique**
- ρ_s la **densité de charge surfacique**.

Ces relations de passages peuvent se déduire des équations de Maxwell précédentes en utilisant la théorie des distributions développée par Laurent Schwartz.

1.1.1.2 Hypothèses pour la suite

Dans la suite de ce mémoire de thèse, nous considérerons, en plus de la continuité, plusieurs hypothèses liées aux matériaux :

1. Les matériaux ne contiennent aucune charge électrostatique localisée. L'équation de Maxwell-Gauss 1.1 devient :
$$\nabla \cdot \mathbf{D} = 0$$
2. Les milieux considérés sont **linéaires**, **locaux** et **isotropes**[1]. On a alors :
$$\mathbf{D}(\mathbf{r}) = \epsilon(\mathbf{r})\mathbf{E}(\mathbf{r}) \tag{1.5}$$

[1] Plus précisément :
- la linéarité signifie que l'induction électrique $\mathbf{D}$ dépend linéairement de $\mathbf{E}$
- la localité signifie que le champ électrique en un point ne dépend des propriétés du matériau qu'en ce point, donc que l'opérateur *produit* à la droite de l'équation 1.5 est bien un produit simple (plutôt qu'un produit de convolution).
- l'isotropie signifie qu'il n'existe pas d'axe particulier pour la grandeur $\epsilon(\mathbf{r})$; elle n'est donc pas un tenseur mais un scalaire.

La grandeur scalaire $\epsilon(\mathbf{r})$ est la **permittivité diélectrique** du milieu ; elle s'exprime en Farad par mètre ($F.m^{-1}$).

3. Les matériaux sont **amagnétiques**, c'est-à-dire qu'ils n'exercent aucune influence sur d'éventuelles lignes de champ magnétique ; alors la **perméabilité magnétique relative** devient unitaire ($\mu_r = 1$) et l'induction magnétique $\mathbf{B}$ s'exprime simplement en fonction du vecteur champ magnétique $\mathbf{H}$:

$$\mathbf{B} = \mu_0 \mu_r \mathbf{H}$$

La constante magnétique μ_0 est la perméabilité magnétique du vide. Elle vaut :

$$\mu_0 = 4\pi 10^{-7} H.m^{-1}$$

4. Les matériaux, s'ils sont métalliques, suivent localement la **loi d'Ohm**. Ils sont alors dits "ohmiques" et cela signifie que le vecteur densité de courant s'exprime en fonction de la **conductivité spécifique** $\sigma(\mathbf{r})$. Celle-ci est scalaire ici, car le matériau est isotrope.

$$\mathbf{J}(\mathbf{r}) = \sigma(\mathbf{r})\mathbf{E}(\mathbf{r})$$

5. La densité de courant surfacique aux interfaces $\mathbf{J}_s$ est nulle. On considère que le courant traversant un matériau est *surfacique* s'il est concentré sur une profondeur δ faible par rapport à l'épaisseur du milieu. La profondeur de pénétration (donnée par $\delta = \sqrt{\frac{2}{\omega\mu\sigma}}$) des ondes visibles (de longueur d'onde de l'ordre de $\lambda = 600nm$) dépasse la centaine de nanomètres dans les résines les plus absorbantes mais n'est par exemple que de $1nm$ pour un métal comme le cuivre[2]

Finalement, les équations locales de Maxwell à considérer seront :

$$\begin{cases} \nabla \cdot \mathbf{E} = 0 & (1.6) \\ \nabla \times \mathbf{E} = -\mu_0 \dfrac{\partial \mathbf{H}}{\partial t} & (1.7) \\ \nabla \cdot \mathbf{H} = 0 & (1.8) \\ \nabla \times \mathbf{H} = \sigma \mathbf{E} + \epsilon \dfrac{\partial \mathbf{E}}{\partial t} & (1.9) \end{cases}$$

et les relations de passages correspondantes (on n'utilisera dans la suite que les relations correspondant à la **continuité des composantes tangentielles** des champs) :

$$\begin{cases} \mathbf{n}_{12} \times (\mathbf{E}_2 - \mathbf{E}_1) = 0 & (1.10) \\ \mathbf{n}_{12} \times (\mathbf{H}_2 - \mathbf{H}_1) = 0 & (1.11) \end{cases}$$

1.1.2 L'onde lumineuse dans le vide

On se place ici dans le vide ou dans un matériau similaire, comme l'air. Alors la conductivité σ est nulle et on peut déduire du système d'équations de Maxwell l'équation de d'Alembert suivante [3] :

[2] Cette hypothèse est notamment lourde de conséquences pour le fonctionnement de la MMFE (cf. annexe B) : les couches métalliques n'autorisent pas l'hypothèse de courants surfaciques nuls, et cette hypothèse est pourtant au coeur de la méthode.

[3] Pour cela on combinera 1.7 et 1.9 de manière à éliminer $\mathbf{E}$ ou $\mathbf{H}$. On utilisera ensuite l'identité vectorielle

$$\nabla^2 \mathbf{A} = \nabla(\nabla \cdot \mathbf{A}) - \nabla \times \nabla \mathbf{A}$$

$$\left(\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)\mathbf{U} = 0 \tag{1.12}$$

avec $1/c^2 = \epsilon_0\mu_0$, c étant la célérité de la lumière dans le vide. Cette équation est valable pour le champ électrique ($\mathbf{U} = \mathbf{E}$) comme pour le champ magnétique ($\mathbf{U} = \mathbf{H}$). C'est **l'équation de propagation de la lumière dans le vide** ; elle a notamment comme solution toute combinaison d'ondes planes progressives harmoniques et monochromatiques.

Nous considérerons donc par la suite que le rayonnement incident s'exprime comme une combinaison d'ondes élémentaires de la forme[4][76] :

$$\mathbf{\Psi}(\mathbf{r}, t) = \mathbf{\Psi}_0 e^{i(\omega t - \mathbf{k}\cdot\mathbf{r})} \tag{1.13}$$

où ω est la pulsation (en s^{-1}) et $\mathbf{k}$ le vecteur d'onde indiquant la direction de propagation. $\mathbf{\Psi}$ représente ici n'importe quelle grandeur vectorielle associée à la propagation électromagnétique : $\mathbf{E}$, $\mathbf{H}$ ainsi que les inductions.

Dans ce manuscrit, il sera toujours question d'un système (un échantillon de matière) excité par un rayon de lumière provenant du vide (ou de l'air). Les grandeurs physiques varieront donc de manière harmonique, en accord avec la longueur d'onde de la lumière. L'opérateur dérivée temporelle devient ainsi, en notation complexe :

$$\frac{\partial}{\partial t} = i\omega t$$

Et l'équation 1.12 devient (avec $k_0 = \omega/c$ le nombre d'onde dans le vide) :

$$\left(\nabla^2 + k_0^2\right)\mathbf{U} = 0 \tag{1.14}$$

1.1.3 L'onde lumineuse dans les matériaux continus

L'excitation harmonique provenant du vide va se propager dans d'autres types de matériaux continus et provoquer là aussi la propagation de phénomènes harmoniques. Le changement d'expression de la dérivée temporelle due à l'excitation harmonique transforme les équations de Maxwell de cette façon :

$$\begin{cases} \nabla\cdot\mathbf{E} = 0 \\ \nabla\times\mathbf{E} = -i\omega\mu_0\mathbf{H} \\ \nabla\cdot\mathbf{H} = 0 \\ \nabla\times\mathbf{H} = \sigma\mathbf{E} + i\omega\epsilon\mathbf{E} \end{cases}$$

On définit, de manière générale, la **permittivité diélectrique relative complexe** $\tilde{\epsilon_r}(\mathbf{r})$ par :

$$\epsilon_0\tilde{\epsilon_r}(\mathbf{r}) = \epsilon(\mathbf{r}) - \frac{i}{\omega}\sigma(\mathbf{r}) \tag{1.15}$$

Il vient ainsi ce système d'équations de Maxwell simplifié mettant en jeu de manière analogue les champs électrique et magnétique :

puis l'une ou l'autre des divergences nulles (1.6 ou 1.8).

[4] Ceci n'est vrai que loin de la source du rayonnement. Nous considérerons donc que celle-ci est située à l'infini (Il s'agit des conditions de Fraunhoffer). L'utilisation dans l'expression du produit scalaire $\mathbf{k}\cdot\mathbf{r}$ est justifiée par le fait que le front d'onde, à l'infini, est quasiment plan.

$$\begin{cases} \nabla \cdot \mathbf{E} = 0 & (1.16) \\ \nabla \times \mathbf{E} = -i\omega\mu_0 \mathbf{H} & (1.17) \\ \nabla \cdot \mathbf{H} = 0 & (1.18) \\ \nabla \times \mathbf{H} = +i\omega\epsilon_0\tilde{\epsilon}_r \mathbf{E} & (1.19) \end{cases}$$

Ainsi que l'équation de propagation (équation d'Helmholtz) :

$$\left(\nabla^2 + k_0^2 \tilde{\epsilon}_r\right) \mathbf{U} = 0 \quad (1.20)$$

Le fait d'inclure le comportement ohmique en régime harmonique dans l'expression de la permittivité complexe (cf. éq. 1.15) permet de décrire de manière unifiée l'évolution d'une onde électromagnétique dans des matériaux diélectriques et dans des métaux ohmiques. Cela s'avère nécessaire car, dans la plupart des cas, un comportement ohmique apparaît dans les résines étudiées pour des longueurs d'ondes incidentes ultra-violettes (cf. partie IV).

Dans la suite, nous utiliserons toujours la permittivité diélectrique relative complexe $\tilde{\epsilon}_r$, et par souci de commodité, nous la noterons parfois simplement ϵ.

1.1.4 Polarisation de la lumière

L'onde plane progressive qui se propage dans le vide (éq. 1.13) est telle que le trièdre $(\mathbf{k}, \mathbf{E}, \mathbf{H})$ est direct. Ceci implique que les champs $\mathbf{E}$ et $\mathbf{H}$ sont perpendiculaires l'un par rapport à l'autre et sont tous les deux situés dans le plan perpendiculaire à la direction d'incidence de l'onde, donnée par $\mathbf{k}$ (cf. fig. 1.3).

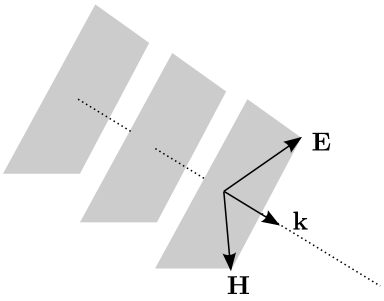


Fig. 1.3: Onde plane progressive harmonique

On dit qu'une lumière est polarisée si l'un des champs varie de manière non aléatoire dans le plan d'onde (perpendiculaire à $\mathbf{k}$). En pratique, l'usage de polariseurs fait que, lors des manipulations, le cas général d'une lumière polarisée est une ellipse ; les cas particuliers étant naturellement les polarisations circulaires et rectilignes comme il est illustré sur la figure 1.4.

Toute onde plane progressive, de polarisation quelconque, arrivant sur une surface plane peut voir ses composantes différenciées. Comme il est explicité sur la figure 1.5, le champ électrique est la somme d'une partie parallèle au plan d'incidence (celui-ci étant le plan perpendiculaire à la surface de l'échantillon et contenant le vecteur d'onde $\mathbf{k}$), $\mathbf{E}_p$, et d'une partie perpendiculaire au plan d'incidence, $\mathbf{E}_s$ (de *senkrecht*, perpendiculaire en allemand). Ces deux composantes perpendiculaires entre elles permettent de traiter indépendamment deux cas remarquables :

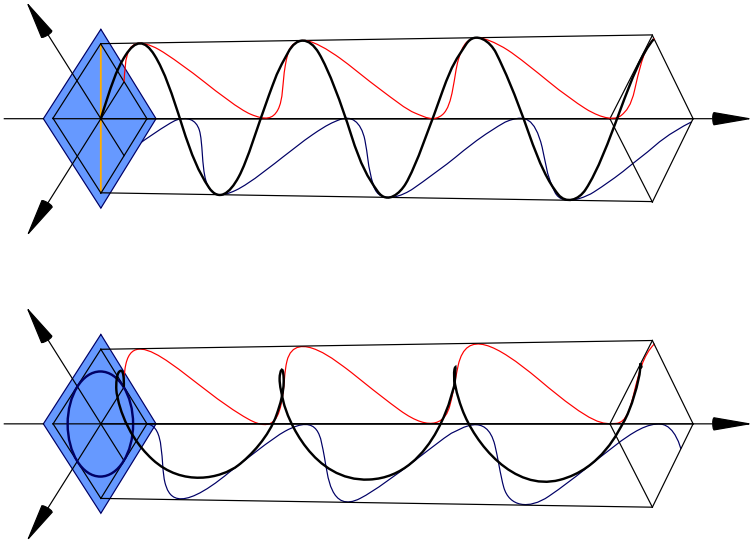


Fig. 1.4: Polarisation rectiligne pour la figure du haut, et polarisation circulaire (cas particulier d'elliptique) pour celle du bas, d'une onde plane. Les tracés bleus et rouges sont les projections du champ électrique (ou magnétique) sur des axes perpendiculaires.

- le cas **transverse électrique** (TE) correspond à un champ électrique perpendiculaire au plan d'incidence : $\mathbf{E}_s$
- le cas **transverse magnétique** (TM) correspond à un champ magnétique perpendiculaire au plan d'incidence, donc un champ électrique parallèle à ce plan : $\mathbf{E}_p$

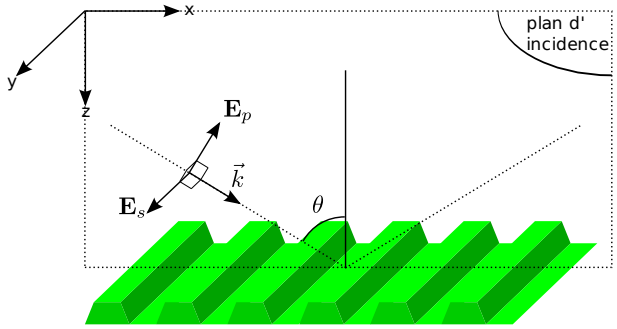


Fig. 1.5: Composantes parallèle ($\mathbf{E}_p$) et perpendiculaire ($\mathbf{E}_s$) du champ électrique par rapport au plan d'incidence

Ces deux cas seront souvent appelés *mode* TE, ou *mode* TM. Il s'agit à chaque fois de polarisation rectiligne, mais dont l'axe est remarquable (remarquable car parallèle à la surface étudiée pour TE ou au plan d'incidence pour TM).

Dans la suite, on distinguera les cas TE et TM, même lorsqu'il ne s'agit plus vraiment d'ondes planes progressives, comme par exemple lors de la propagation du champ électromagnétique dans une région inhomogène ; ces cas se réfèreront en fait à la polarisation TE ou TM de l'onde incidente provenant du vide, celle-là même qui excite le système harmoniquement.

1.2 Indices des matériaux

On suppose ici qu'une onde plane progressive se propage dans un milieu continu. On introduit la forme de cette onde plane 1.13 dans l'équation de propagation 1.20. Il vient :

$$-k^2 + k_0^2 \tilde{\epsilon_r} = 0$$

De là, on exprime la **vitesse de phase** $v_\phi = \omega/k$:

$$\left(\frac{1}{v_\phi}\right)^2 = \frac{k^2}{\omega^2} = \frac{k_0^2 \tilde{\epsilon_r}}{\omega^2} = \frac{1}{c^2}\tilde{\epsilon_r}$$

L'**indice de réfraction complexe** ou **indice optique** d'un matériau est défini comme le rapport de la célérité de la lumière dans le vide par la vitesse de phase de l'onde dans le milieu. Ce rapport (au carré) vaut ici :

$$\bar{n} = \frac{c}{v_\phi} = \sqrt{\tilde{\epsilon_r}} = \eta - i\kappa \tag{1.21}$$

Ainsi, l'onde se propageant dans le milieu (cf. équation 1.13) s'exprime :

$$\Psi(\mathbf{r}, t) = \Psi_0 e^{-\kappa \frac{\omega}{c} \frac{\mathbf{k}\cdot\mathbf{r}}{||\mathbf{k}||}} e^{i\left(\omega t - \eta \frac{\omega}{c} \frac{\mathbf{k}\cdot\mathbf{r}}{||\mathbf{k}||}\right)}$$

- η est appelé **coefficient de dispersion**, *indice de réfraction réel* ou *indice optique réel*. Il traduit le fait que les différentes longueurs d'ondes ne traversent pas le matériau à la même vitesse. Il est aussi responsable du changement de direction d'un rayon lumineux au passage d'un dioptre (loi de Descartes).
- κ est le **coefficient d'extinction** ou d'atténuation. Il est relatif à la perte d'énergie électromagnétique d'un rayon lumineux à la traversée du matériau. Cette perte peut être due aux phénomènes d'absorption, de diffusion (si le milieu n'est pas continu) ou de luminescence. Les deux derniers phénomènes ne sont pas modélisables par la simple électrodynamique des milieux continus utilisée ici.

Dans la suite de ce mémoire de thèse, et dans la mesure où cela n'induit pas de confusion (avec le nombre d'onde k notamment), nous utiliserons les symboles (n, k) plutôt que (η, κ) ; ceci afin de mieux correspondre aux notations habituelles.

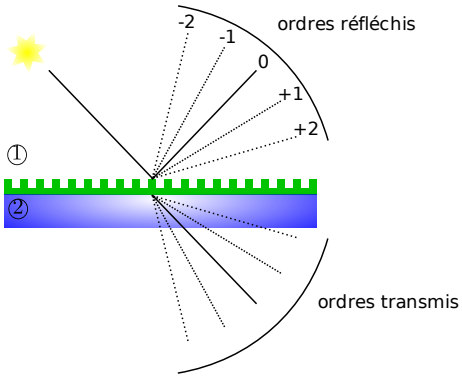
1.3 Diffraction par un réseau

Le phénomène de diffraction est une conséquence directe de la nature ondulatoire de la lumière : cette caratéristique provoque des interférences qui peuvent être **constructives** lorsque deux ondes additionnent localement leur amplitude complexe, ou **destructives** lorsque leurs amplitudes s'annihilent l'une l'autre.

Ce phénomène se révèle lors de l'interaction d'une onde lumineuse avec un objet présentant des variations géométriques dont la dimension caractéristique est de l'ordre de grandeur de la longueur d'onde incidente.

On peut ainsi, par exemple, voir diffracter de la lumière en illuminant un réseau périodique. Le rayon incident semblera alors se diviser (se diffracter) en de multiples ordres réfléchis et transmis, comme cela est représenté sur la figure 1.6(a). Ces ordres sont affectés à des angles qui dépendent de

la longueur d'onde incidente. Ainsi, si on utilise de la lumière naturelle (blanche, contenant toutes les longueurs d'ondes), on verra apparaître un reflet irisé (les couleurs sont séparées). Sur la photographie 1.6(b) est représenté un disque compact qui, en coupe, est un réseau de diffraction de période $2\mu m$.



(a) Ordres réfléchis et transmis pour une longueur d'onde

(b) Disque compact

Fig. 1.6: Phénomène de diffraction

Une des représentations théoriques de ce phénomène est donnée par le **principe de Huygens-Fresnel**. Ce principe dit que l'interaction d'un rayon lumineux avec le milieu matériel diffractant génère en chaque point de la surface une onde sphérique qui reprend l'amplitude et la phase de l'onde incidente. Ce sont ces multiples ondes sphériques qui vont interférer pour constituer la figure de diffraction.

Il convient de noter enfin que cette représentation suppose une distance d'observation grande par rapport aux dimensions du système diffractant.

Formule des réseaux. Les angles des différents ordres réfléchis et transmis sont déterminés en fonction des interférences constructives des ondes réémises par le milieu qui subit l'excitation. Sur le schéma 1.7 on a représenté les différences de phases entre deux rayons ① et ② subissant une réflexion ou une transmission à une distance de Λ_x correspondant à la période du réseau.

Pour obtenir une interférence constructive, il est nécessaire que cette différence de phase,

$$\delta = \delta_{r,t} - \delta_i$$

soit multiple de 2π. Dans ce cas, la composante tangentielle du vecteur d'onde de l'ordre m diffracté (qu'il soit transmis ou réfléchi) est :

$$\alpha_m = (k_x)_m = k_{r,t} \sin(\theta_{r,t})_m = \tilde{n}_0 k_0 \sin\theta_i + mK_x \tag{1.22}$$

où $k_{r,t}$ est le nombre d'onde de la lumière dans le milieu de réflexion ou de transmission, $(\theta_{r,t})_m$ l'angle de l'ordre diffracté m dans ces milieux, $\tilde{n}_0$ l'indice optique complexe du milieu d'incidence, θ_i l'angle d'incidence et K_x est le nombre d'onde du réseau ($K_x = 2\pi/\Lambda_x$).

La justification géométrique de la formule des réseaux fait appel aux indices optiques réels. Nous avons généralisé aux grandeurs complexes car les relations restent valables quelle que soit la valeur du coefficient d'extinction, celui-ci n'ayant aucun effet sur la phase de l'onde.

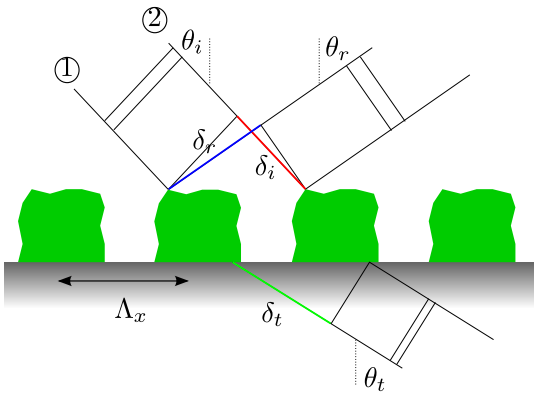


Fig. 1.7: Formule des réseaux de diffraction

CHAPITRE 2

Ellipsométrie et scatterométrie

La réponse optique d'un échantillon sur lequel on projette une onde lumineuse est significative de sa nature, notamment de l'indice optique $\tilde{n}$ des matériaux le constituant et de la répartition géométrique de ces matériaux.

Ainsi, si l'on connaît parfaitement cette répartition, s'il s'agit par exemple d'un matériau unique massif et plan ou bien de matériaux déposés en couches planes, alors il est possible de retrouver les indices optiques (lire la partie IV à ce sujet).

D'autre part, si l'on connaît parfaitement les indices optiques, il est possible d'acquérir de l'information sur la géométrie du système. C'est sur ce principe que se base la **scatterométrie** : si l'on possède *a priori* l'information que l'échantillon est **périodique** (de période suffisamment faible pour faire diffracter la lumière), et qu'il est constitué de matériaux connus, alors il est possible d'acquérir un supplément d'information sur la répartition des matériaux en analysant la réponse optique.

On peut prendre l'exemple d'un système diffractant constitué d'un créneau de résine disposé sur un substrat plan de silicium (fig. 2.1) ; la scatterométrie permet de déterminer les grandeurs géométriques H et CD du profil préalablement modélisé.

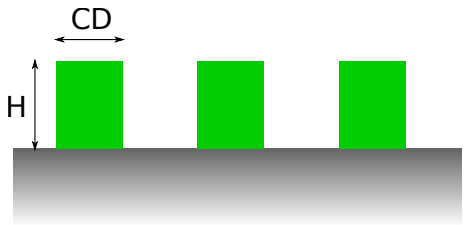


Fig. 2.1: La scatterométrie utilise la réponse optique d'un profil diffractant pour déterminer ses paramètres géométriques : ici, une largeur (CD) et une hauteur de créneau (H).

Le mot scatterométrie est un néologisme venant de l'anglais *scatterometry*, de *scatter* signifiant diffracter. Un mot français plus correct pourrait être diffractométrie. Ne désignant que l'objet d'étude (la lumière diffractée) et pas la méthode de mesure, la scatterométrie peut être *réflectométrique* si l'on considère le changement d'intensité de lumière provoqué par la diffraction ou encore *ellipsométrique* si l'on considère le changement de polarisation. Nous avons choisi, en fonction notamment du matériel

expérimental dont nous disposions, d'utiliser des mesures ellipsométriques.

2.1 Principe de l'ellipsométrie

Le principe de base de l'ellipsométrie est de mesurer le changement d'état de polarisation de la lumière induit par la réflexion sur une surface ; le plus souvent (si la réflexion n'induit pas de dépolarisation), une polarisation *linéaire* (cas particulier) deviendra *elliptique* (cas général). Comme on l'a vu en 1.1.4, le champ électrique incident $\mathbf{E}^i$ peut se décomposer en une composante parallèle (p) et une composante perpendiculaire (s) au plan d'incidence. (cf. figure 2.2)

$$\mathbf{E}^i = \mathbf{E}^i_p + \mathbf{E}^i_s$$

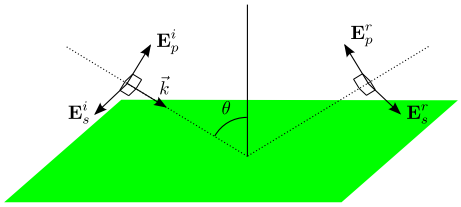


Fig. 2.2: Changement de polarisation à la réflexion

L'ellipsométrie consiste donc à mesurer la grandeur :

$$\rho = \tan(\Psi)e^{i\Delta} = \frac{r_p}{r_s}$$

où r_p et r_s sont les rapports (respectivement, pour les composantes p et s) des amplitudes des champs incidents et réfléchis :

$$r_p = \frac{\mathbf{E}^r_p}{\mathbf{E}^i_p} \qquad r_s = \frac{\mathbf{E}^r_s}{\mathbf{E}^i_s} \tag{2.1}$$

Cette grandeur ρ est non seulement dépendante de l'échantillon visé par le rayon lumineux, mais aussi des conditions expérimentales, en particulier de l'angle d'incidence θ et de la longueur d'onde incidente λ. Une caractérisation de l'objet par ellipsométrie se fera donc en faisant varier, sur une plage connue, l'une de ces variables. On parle alors :

- d'**ellipsométrie spectroscopique**, si l'on fait varier la longueur d'onde λ (dans ce cas, l'angle θ est constant) ;
- d'**ellipsométrie à angle variable**, ou ellipsométrie 2θ [44] ou encore goniométrie (quoique ce terme s'utilise généralement pour des mesures réflectométriques) si l'on fait varier à l'aide d'un bras motorisé l'angle θ.

Dans le cas de l'ellipsométrie spectroscopique en particulier, on parle de **signature** pour désigner les deux courbes issues de la mesure : $\Psi(\lambda)$ et $\Delta(\lambda)$ par exemple. De nombreux exemples de signatures seront donnés par la suite.

2.2 L'ellipsomètre

L'**ellipsomètre** est constitué au minimum d'une source de lumière monochromatique (polarisée circulairement), d'un polariseur (chargé de polariser le rayon linéairement), d'un analyseur puis d'un détecteur (par exemple un photomultiplicateur).

Ellipsomètre à extinction. Parmi les nombreux systèmes existants, celui-ci est le plus simple. Schématisé sur la figure 2.3, il est construit à partir de la configuration de base à laquelle on a ajouté un compensateur en aval du polariseur. En le tournant d'un angle C, celui-ci permet d'obtenir une polarisation rectiligne après la réflexion sur l'échantillon. En tournant l'analyseur d'un angle A correspondant à une orientation perpendiculaire à celle de la polarisation, le signal à l'entrée du détecteur s'annule.

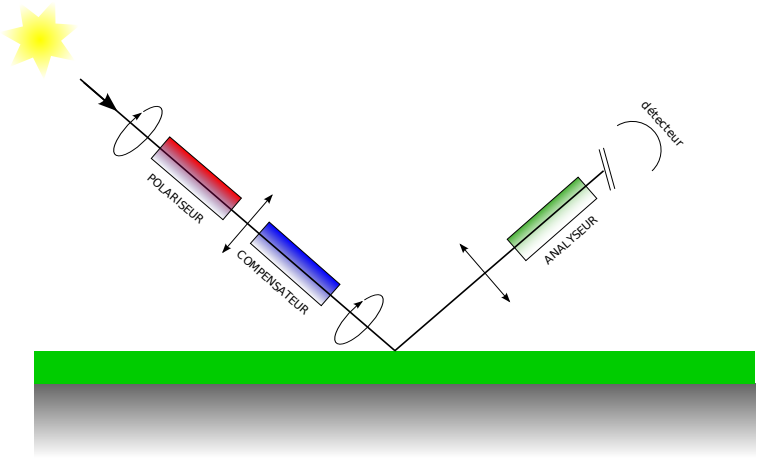


Fig. 2.3: Ellipsomètre à extinction

On peut ainsi remonter aux grandeurs ellipsométriques :

$$\rho = \tan\Psi e^{j\Delta} = -\tan A \frac{\tan C - \tan(P-C)}{1 + i\tan C\tan(P-C)} \tag{2.2}$$

où P, C et A sont les angles donnés respectivement au polariseur, au compensateur et à l'analyseur.

Cette méthode donne accès directement aux valeurs Ψ et Δ mais nécessite, en cas d'automatisation, une étape lente et délicate de recherche d'intensité minimale. Cette étape est d'autant plus délicate qu'elle est rendue imprécise par les bruits de fond du détecteur. Nous utiliserons donc un autre appareillage sans pièce mécanique et permettant des vitesses d'acquisition plus élevées :

L'ellipsomètre spectroscopique à modulation de phase. L'idée est d'ajouter, en aval du polariseur, un modulateur qui créera un déphasage périodique sinusoïdal entre les deux composantes p et s du champ électrique.

Ce modulateur est en fait un barreau de verre qui est contraint mécaniquement par un système piézoélectrique (cf. schéma 2.4) ; ceci fait qu'il devient biréfringent, c'est à dire que son indice optique selon l'axe de la contrainte $\tilde{n_1}$ est modulé par rapport à celui de l'axe perpendiculaire $\tilde{n_0}$.

De plus, on ajoute un monochromateur qui permettra de sélectionner une longueur d'onde particulière (ou d'en séparer de multiples) en entrée d'un capteur d'intensité lumineuse. L'intensité reçue est une fonction non seulement de la pulsation de modulation, mais aussi des grandeurs ellipsométriques que l'on cherche à déterminer (section 2.3).

Ce système possède ainsi plusieurs avantages : premièrement, la lumière incidente peut ne pas être polarisée ni monochromatique (ce sont respectivement le polariseur et le monochromateur qui changeront cela) ; deuxièmement, la fréquence d'acquisition des grandeurs ellipsométriques est potentiellement élevée (avec une pulsation $\omega = 50kHz$, on peut effectuer des centaines de mesures

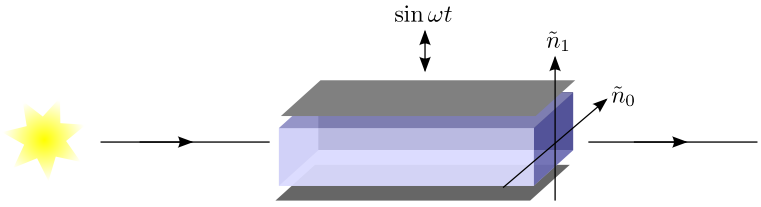


Fig. 2.4: Modulation de l'indice optique par contrainte piézoélectrique

par seconde, mais une bonne précision suppose un temps plus élevé pour moyenner les mesures) ; troisièmement, il est relativement peu sensible aux mouvements parasites des instruments d'optique.

Il impose cependant la contrainte de la calibration du modulateur : celui-ci doit être maintenu à une température contrôlée et la tension aux bornes du piézo-électrique doit être adaptée à la longueur d'onde (ou, du moins, à une plage de longueurs d'ondes).

La figure 2.5 représente un tel ellipsomètre à modulation de phase.

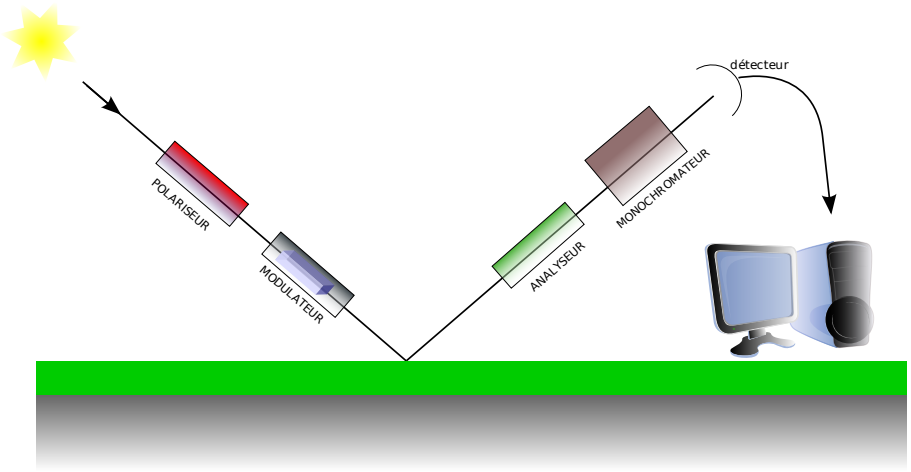


Fig. 2.5: Ellipsomètre à modulation de phase

L'ellipsomètre Jobin-Yvon UVISEL. Au cours de cette thèse, nous avons utilisé deux ellipsomètres : l'un situé sur une chambre de gravure plasma et l'autre sur une plateforme externe munie d'un porte-échantillon chauffant et positionnable. Ces deux ellipsomètres à modulation de phase sont construits par la société Jobin-Yvon et sont constitués notamment :

- d'une source lumineuse au xénon ($\lambda = 190 - 2100nm$) 150W refroidie par azote et munie d'un système fournissant un point lumineux de taille submillimétrique ;
- d'un modulateur photoélastique (50kHz) stabilisé thermiquement ;
- d'un système d'acquisition multi-longueur d'onde MWL32 (chambre plasma) et MWL64 (plateforme externe) muni de photomultiplicateurs permettant de mesurer en parallèle, respectivement 16 et 32 longueurs d'ondes (à choisir dans la gamme 190-925nm). La fréquence d'acquisition maximale donnée par le constructeur est de 20 par seconde (pour le MWL64).

Ce type d'ellipsomètre, décrit en référence [81], est représenté sur la photographie 2.6. Il présente la caractéristique particulière d'avoir le modulateur sur le bras du détecteur plutôt que sur le bras de

la source. Cela change peu de choses dans les équations de la section suivante (données généralement dans la littérature) et en tout état de cause, ne change rien au résultat.

Fig. 2.6: Ellipsomètre Jobin-Yvon UVISEL associé au porte-échantillon chauffant.

2.3 Grandeurs mesurées et déduites

Comme il a été dit précédemment, un système de mesure ellipsométrique fourni de manière générale un couple de signaux pour chaque longueur d'onde. Plusieurs couples sont couramment utilisés et nous les décrirons dans cette section.

2.3.1 Couple (I_S, I_C)

L'intensité lumineuse détectée par l'ellipsomètre est de la forme (cf. [16]) :

$$I(t) = I_0 + I_S \sin\delta(t) + I_C \cos\delta(t) \tag{2.3}$$

où :

$$\delta(t) = a\sin(\omega t) \tag{2.4}$$

est le déphasage sinusoïdal entre les composantes du champ électrique créé par l'élément piézo-électrique (a est une constante à déterminer), et :

$$\begin{cases} I_0 = B(1 - \cos 2\Psi\cos 2A + \cos 2(P-M)\cos 2M(\cos 2A - \cos 2\Psi) \\ \quad + \sin 2A\cos\Delta\cos 2(P-M)\sin 2\Psi\sin 2M) \\ I_S = B(\sin 2(P-M)\sin 2A\sin 2\Psi\sin\Delta) \\ I_C = B(\sin 2(P-M)\,[(\cos 2\Psi - \cos 2A)\sin 2M + \sin 2A\cos 2M\sin 2\Psi\cos\Delta]) \\ B = \dfrac{E_0^2}{4\left|r_p^2 + r_s^2\right|} \end{cases} \tag{2.5}$$

$\delta(t)$ étant elle même une sinusoïde, les fonctions $\sin\delta(t)$ et $\cos\delta(t)$ peuvent être développées en séries de Fourier à l'aide des fonctions de Bessel de première espèces en a, $J_m(a)$ (cf. équation 2.4) :

$$\begin{cases} \sin\delta(t) = 2\sum_{m=0}^{\infty} J_{2m+1}(a)\sin[(2m+1)\omega t] \\ \cos\delta(t) = J_0(a) + 2\sum_{m=1}^{\infty} J_{2m}(a)\sin[(2m)\omega t] \end{cases}$$

En pratique, on ne considère que les harmoniques d'ordres 0,1 et 2, alors on a :

$$I(t) = I_0 + I_S[2J_1(a)\sin(\omega t)] + I_C[J_0(a) + 2J_2(a)\cos(2\omega t)]$$

On s'arrange pour annuler $J_0(a)$, c'est à dire fixer $a = 2.405$. Ceci se fait préalablement à toute mesure ; en effet le déphasage $\delta(t)$ induit par le barreau biréfringent est fonction de la tension V aux bornes de l'élément piézoélectrique (Q est une constante propre à celui-ci) :

$$\delta(t) = a\sin(\omega t) = Q\frac{V}{\lambda}\sin(\omega t)$$

Ainsi, pour garder a constant il faudra fournir une tension V différente pour chaque longueur d'onde λ, d'où la difficulté à calibrer ce type d'appareil.

Au final, on se retrouve avec une intensité lumineuse modélisée de la forme :

$$I = I_0(R_\omega \sin(\omega t) + R_{2\omega}\cos(2\omega t)) \tag{2.6}$$

avec :

$$\begin{cases} R_\omega = 2J_1(a)\dfrac{I_S}{I_0} \\ R_{2\omega} = 2J_2(a)\dfrac{I_C}{I_0} \end{cases}$$

Pour obtenir les grandeurs I_S et I_C, on devra donc démoduler l'intensité reçue par l'ellipsomètre et en retirer les deux harmoniques R_ω et $R_{2\omega}$.

2.3.2 Couple (Ψ, Δ)

Afin de déterminer de manière unique les grandeurs ellipsométriques Ψ et Δ, on configure les angles d'orientation du polariseur, du modulateur et de l'analyseur de deux manières distinctes telles que l'intensité non modulée (éq. 2.5) soit unitaire : $I_0 = 1$

– **Configuration I** : $(P - M) = \pi/4, M = 0, A = \pi/4$ alors :

$$\begin{cases} I_S &= \sin 2\Psi \sin\Delta \\ I_C &= \sin 2\Psi \cos\Delta \end{cases}$$

donc (cf. fig. 2.7) :

$$\sin^2(2\Psi) = I_S^2 + I_C^2$$
$$\sin(2\Psi) = +\sqrt{I_S^2 + I_C^2}$$

On rejette la solution $\sin(2\Psi) = -\sqrt{I_S^2 + I_C^2}$ car

$$tan(\Psi) \geq 0 \Rightarrow \Psi \in [0; \pi/2] \Rightarrow \sin(2\Psi) \geq 0$$

Néanmoins, la valeur de Ψ est double.
En revanche, la valeur de Δ est unique car choisie en fonction du signe de I_S , $sgn(\sin\Delta) = sgn(I_S)$:

$$tan(\Delta) = I_S/I_C$$

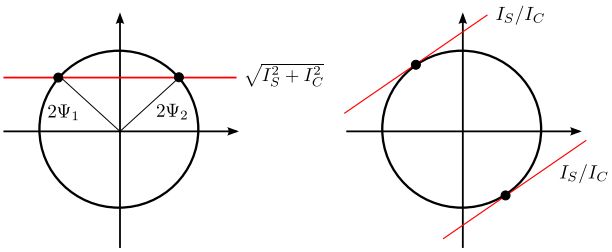


Fig. 2.7: Représentation des angles Ψ (à gauche) et Δ (à droite) pour la configuration I

– **Configuration II** : $(P - M) = \pi/4, M = \pi/4, A = \pi/4$, alors :

$$\begin{cases} I_S = \sin 2\Psi \sin\Delta & (2.7) \\ I_C = \cos 2\Psi & (2.8) \end{cases}$$

donc (cf. fig. 2.8) :

$$\cos(2\Psi) = I_C$$

La solution est unique car le signe de $\sin(2\Psi)$ est connu positif. D'autre part :

$$\sin^2(\Delta) = -I_S/I_C$$
$$\sin(\Delta) = \pm\sqrt{-I_S/I_C}$$

On choisi le signe $\pm\sqrt{-I_S/I_C}$ en fonction du signe de I_S, car $sgn(\sin\Delta) = sgn(I_S)$; dans les deux cas, la solution est double.

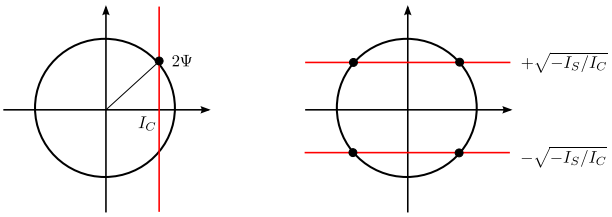


Fig. 2.8: Représentation des angles Ψ (à gauche) et Δ (à droite) pour la configuration II

En conclusion, si l'on cherche à déterminer de manière unique le couple (Ψ, Δ), il faudra opérer une mesure dans chacune des configurations I et II. Cependant, si l'on cherche seulement à comparer une mesure à des grandeurs calculées, on choisira d'utiliser le couple (I_S, I_C) qui est déterminé de manière unique par une seule des configurations.

Les domaines de définition des grandeurs Ψ et Δ sont théoriquement, selon ce qu'on vient de voir :

$$\Psi \in [0; \pi/2]$$
$$\Delta \in [0; 2\pi]$$

Néanmoins, on peut extraire du signal mesuré I (cf. équation 2.6) aussi bien le couple de valeurs (I_S, I_C), que le couple $(-I_S, -I_C)$ (mathématiquement, il s'agit d'un décalage de phase). Il est donc naturel de convenir du signe de l'une des deux valeurs : pour les ellipsomètres KLA-Tencor par exemple, il est convenu que $I_S \geq 0$ (donc $\sin \Delta \geq 0$), et pour ceux de fabrication Jobin-Yvon au contraire $I_C \geq 0$ (donc $\Psi \in [0; \pi/4]$). La figure 2.9 illustre ces conventions.

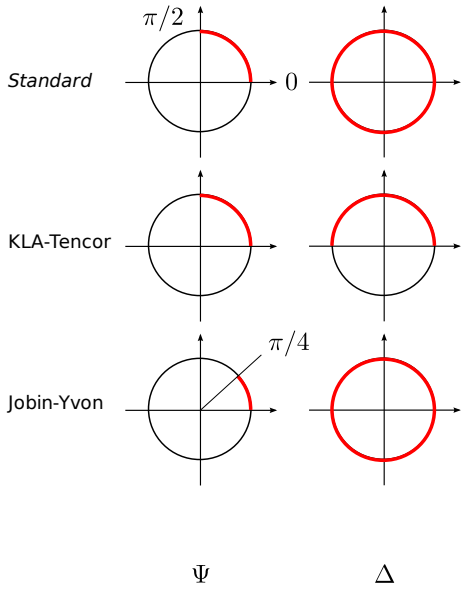


Fig. 2.9: Ensembles de définition des grandeurs Ψ et Δ

2.4 Une autre méthode de caractérisation : la réflectométrie spectroscopique

Un système de réflectométrie spectroscopique mesure le coefficient de réflexion R d'un échantillon en fonction de la longueur d'onde incidente. Il est constitué d'une source et d'un système optique qui permet de comparer l'intensité de la lumière envoyée avec un angle d'incidence θ avec l'intensité réfléchie. S'il y a diffraction, alors la mesure concernera seulement l'ordre 0. Lorsqu'on adjoint à ce système un polariseur, alors on peut mesurer alors les grandeurs réelles R_p et R_s :

- R_p si l'on polarise la lumière perpendiculairement aux lignes du réseau (selon x, mode TM)
- R_s si l'on polarise la lumière parallèlement aux lignes du réseau (selon y, mode TE)

Le schéma de principe d'un exemple de réflectomètre est donné sur la figure 2.10 ; il s'agit d'un réflectomètre à incidence normale ($\theta = 0$) [39]. Ce système ne considère pas le déphasage induit par la réflexion sur l'échantillon ; il fournit donc les réels R_p et R_s plutôt que deux complexes (Ψ et Δ), mais sa simplicité et sa compacité en font un outil très utilisé dans l'industrie microélectronique pour la métrologie *in situ*.

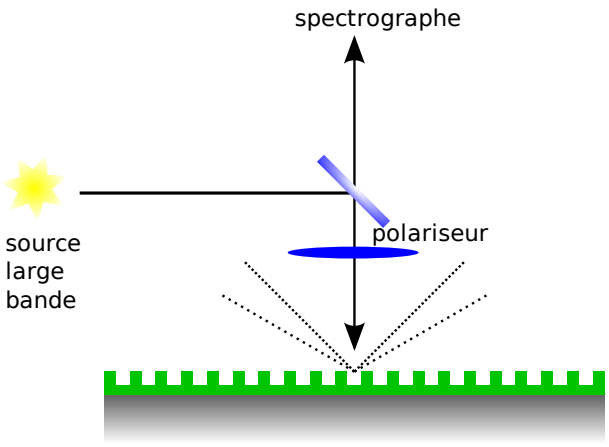


Fig. 2.10: Principe d'un réflectomètre à incidence normale, de type Optical CD(Nanometrics)

CHAPITRE 3

Simulation numérique des réponses optiques

Comme il a été dit au chapitre 2, l'ellipsométrie spectroscopique est une technique de métrologie qui utilise la réponse optique (changement de polarisation) d'un échantillon à l'excitation par un rayon lumineux. Ce chapitre est consacré à la simulation numérique de cette réponse.

Trois problèmes serons considérés et exposés dans l'ordre du plus simple au plus complexe :

- la première section traitera du cas d'un échantillon constitué d'un unique matériau massif (ou semi-infini) ;
- dans la deuxième section, l'échantillon considéré sera constitué de plusieurs matériaux disposés en couches ;
- enfin, nous traiterons le cas plus complexe d'un réseau de lignes périodiques. Les matériaux en jeu sont multiples et la diffraction engendre plusieurs ordres réfléchis, chacun ayant sa propre réponse.

3.1 Matériau massif

On considère ici le cas le plus simple. Il s'agit de calculer le changement de polarisation induit par la réflexion d'une onde sur un matériau. Comme représenté sur la figure 3.1, l'onde provient d'un milieu d'indice $\tilde{n}_0$ avec un angle d'incidence θ_0 et l'échantillon est caractérisé par son indice optique $\tilde{n}_1$.

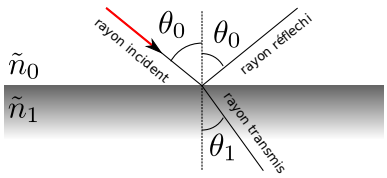


Fig. 3.1: Réflexion sur un matériau semi-infini

La loi de Snell-Descartes dit alors que le rayon est réfléchi avec le même angle θ_0, mais transmis avec un angle θ_1 tel que :

$$n_0 \sin\theta_0 = n_1 \sin\theta_1 \tag{3.1}$$

où n_0 et n_1 sont les indices réels de réfraction des matériaux considérés.

On démontre alors[1] que les grandeurs ellipsométriques r_p et r_s (coefficients de réflexion, cf. éq. 2.1) valent :

$$\begin{cases} r_p = \dfrac{\tilde{n}_1 \cos\theta_0 - \tilde{n}_0 \cos\theta_1}{\tilde{n}_1 \cos\theta_0 + \tilde{n}_0 \cos\theta_1} & (3.2) \\ r_s = \dfrac{\tilde{n}_0 \cos\theta_0 - \tilde{n}_1 \cos\theta_1}{\tilde{n}_0 \cos\theta_0 + \tilde{n}_1 \cos\theta_1} & (3.3) \end{cases}$$

3.2 Ellipsométrie de couches minces

Nous supposons ici qu'une couche mince (par rapport à la longueur de cohérence de la source lumineuse) de matériau est déposée sur un substrat semi-infini. Alors, à l'amplitude des champs dans le demi-plan de la réflexion s'ajoute, par rapport au matériau semi-infini, les amplitudes des champs issus des multiples réflexions à l'intérieur de la couche de matériau. Pour cette raison, l'ellipsométrie mono-couche sur substrat est une technique de mesure extrêmement sensible ; elle est donc particulièrement adaptée à la mesure précise d'épaisseurs de couches minces (ou d'indices optiques, cf. partie IV).

Comme il est illustré sur le schéma 3.2, il convient d'ajouter au rayon simplement réfléchi ① les autres rayons multiplement transmis et réfléchis ②, ③, etc.

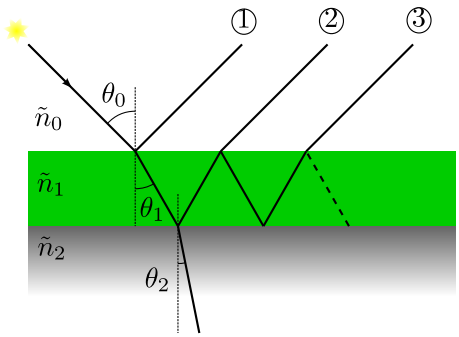


Fig. 3.2: Couche mince de matériaux sur substrat

Le déphasage induit par l'aller-retour de la lumière dans la couche est, pour une longueur d'onde λ et une couche d'épaisseur e_1 et d'indice optique $\tilde{n}_1$:

$$\beta = \frac{4\pi}{\lambda} e_1 \tilde{n}_1 \cos\theta_1$$

[1] Cette démonstration ne sera pas détaillée ici. Elle consiste à écrire les relations de passage à l'interface pour les composantes tangentielles des champs électrique et magnétique (elles sont continues) pour le mode TE puis TM. On démontre en même temps que les coefficients de transmission sont :

$$\begin{cases} t_p = \dfrac{2\tilde{n}_0 \cos\theta_1}{\tilde{n}_1 \cos\theta_0 + \tilde{n}_0 \cos\theta_1} \\ t_s = \dfrac{2\tilde{n}_1 \cos\theta_1}{\tilde{n}_0 \cos\theta_0 + \tilde{n}_1 \cos\theta_1} \end{cases}$$

Ainsi, on peut écrire le coefficient de réflexion total en fonction des coefficients de réflexion sur les dioptres et de transmission entre dioptres (cette égalité est symbolisée sur la figure 3.3) :

$$\begin{aligned} r &= r_{01} + t_{01}r_{12}e^{-i\beta}t_{10} + t_{01}r_{12}e^{-i\beta}r_{10}r_{12}e^{-i\beta}t_{10} + ... \\ &= r_{01} + t_{01}r_{12}e^{-i\beta}t_{10}(1 + q + q^2 + ...) \end{aligned}$$

où $q = r_{10}r_{12}e^{-i\beta}$. r s'exprime donc avec la somme d'une suite géométrique de raison q.

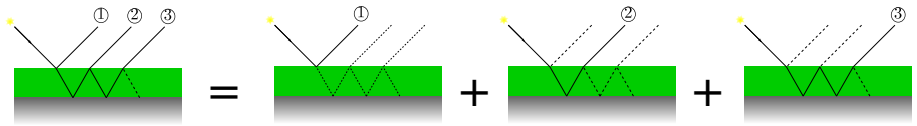


Fig. 3.3: Suite de réflexions et transmissions du rayon lumineux

Et comme $r_{01} = -r_{10}$, il vient $t_{01}t_{10} = (1 - r_{01})(1 + r_{01}) = 1 - r_{01}^2$ et :

$$r = \frac{r_{01} + r_{12}e^{-i\beta}}{1 + r_{01}r_{12}e^{-i\beta}}$$

Les coefficients r_{ij} et t_{ij} s'obtiennent comme il est expliqué dans la section précédente.
Ce calcul de r vaut aussi bien pour les polarisations s et p ; on exprime alors r_s et r_p.

3.3 Réseaux de lignes périodiques

La troisième méthode de simulation présentée dans ce chapitre concerne les échantillons munis d'une surface périodique. Dans notre cas, la période est unidimensionnelle, selon l'axe x et le système est invariant selon y. La surface est donc tapissée de lignes comme le montre la figure 3.4.

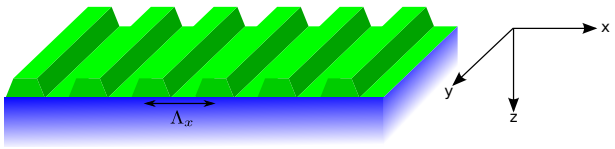


Fig. 3.4: Système diffractant périodique de motif quelconque

Le phénomène de diffraction par un réseau périodique dont le motif est quelconque, en particulier s'il est profond (par rapport à la période du réseau) n'est pas correctement modélisé par le développement en série de Rayleigh. En effet, cette décomposition des champs en simples ondes planes se propageant selon des directions correspondant aux ordres de diffraction (cf. formule des réseaux, 1.3) est valable dans les milieux homogènes mais ne décrit pas correctement la physique à l'intérieur de la région modulée. Elle n'explique notamment pas les anomalies de résonance, dont celle de Wood (résonance plasmonique, [94]).

Pour décrire les efficacités diffractées par un tel réseau, et cela pour les modes TE et TM, il est nécessaire d'utiliser une méthode dite *rigoureuse*, c'est à dire ne faisant appel, formellement, à aucune approximation. Elle devra notamment prendre en compte la nature vectorielle du champ électromagnétique.

Il existe une grande variété de méthodes destinées à simuler ce type de problème (méthode des coordonnées curvilignes [20] [61], méthodes différentielles, intégrales, etc.). Parmi celles-ci, nous avons choisi une méthode fréquemment utilisée en scatterométrie qui consiste à découper la forme du motif diffractant en couches élémentaires dans lesquelles la permittivité diélectrique ϵ est constante selon l'axe vertical z (cf. fig. 3.5).

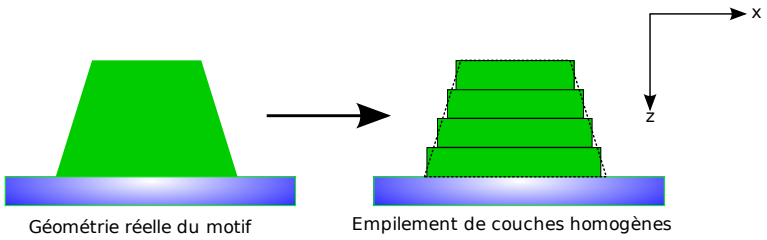


Fig. 3.5: Découpe en couches élémentaires d'un motif diffractant

La nature périodique selon z de chacune des couches nous autorise à développer la permittivité diélectrique $\epsilon(x)$ en série de Fourier et à décomposer les champs dans une base de Floquet (analogue à la décomposition de Rayleigh, mais pour pour les milieux inhomogènes). Le résultat final s'obtient en reliant les modes de propagation issus de cette dernière décomposition par les relations de continuité des composantes tangentielles des champs électriques et magnétiques aux interfaces entre couches.

Formellement, cela consiste à écrire pour chaque couche ces deux développements et à les intégrer aux équations de Maxwell. Cela mène à une suite de problèmes aux valeurs propres dont la résolution fournit, à l'amplitude près, les modes de propagations. Les amplitudes seront déterminées enfin par l'utilisation des relations de passages.

Cette méthode est donc une **méthode modale par développement en série de Fourier**. Elle est communément désignée par les acronymes MMFE (*Modal Method by Fourier Expansion*) ou RCWA (*Rigorous Coupled Waves Analysis*) et sa description détaillée est en annexe B.

Deuxième partie

Problème inverse et reconstruction dynamique de profil

Dans le chapitre 3, il était question du calcul d'une signature scatterométrique (ou du moins, ellipsométrique) produite par un système diffractant connu, c'est-à-dire modélisé géométriquement et constitué de matériaux dont les indices optiques sont connus. Ce problème, qu'il est possible de résoudre en utilisant par exemple une méthode modale (cf. section 3.3 consacrée à la MMFE), est appelé, on l'a vu, *problème direct*.

A contrario, le problème qui consiste à déterminer la nature du système diffractant connaissant la signature scatterométrique est appelé *problème inverse*.

Pour la scatterométrie, les équations et les algorithmes qui permettent de calculer une signature ne sont pas inversibles : il s'agit d'un problème *mal posé*. Cela signifie que la difficulté à résoudre le problème à partir des données disponibles est grande et, en particulier, qu'il est possible qu'une faible variation des données initiales provoque une forte variation dans la solution. Cette caractéristique est due à un *défaut* d'information :

- Premièrement, à cause de la faible largeur du spectre électromagnétique considéré : on l'a vu, les signatures scatterométriques utilisées ne concernent que la partie lumière visible (partie étendue néanmoins $\lambda \in [250; 800] nm$) ;
- Deuxièmement, nos conditions expérimentales n'impliquent que la mesure de l'ordre zéro diffracté : donc pas d'ordres supérieurs, ni propagatifs, ni évanescents (pas de mesures en champ proche).

Les informations habituellement disponibles permettent au mieux de reconstruire un profil diffractant sous forme de gradients d'indices (Elshner *et al.* [82]), ce qui s'avère inexploitable pour une métrologie sensée mesurer avec précision la forme de motifs.

Une manière de résoudre le problème inverse comme nous le souhaitons consiste donc à apporter suffisamment d'information *a priori* dans la description du système diffractant. L'utilisation des mesures scatterométriques permettent ensuite de fixer cette description. En pratique, ces informations *a priori* sont :

- les matériaux utilisés, c'est-à-dire leurs indices optiques sur le spectre de la signature ;
- la géométrie **paramétrée** du système.

Ainsi, tout le problème inverse se ramène à un problème d'**optimisation paramétrique**.

Un exemple typique de problème inverse en scatterométrie est exploité dans le chapitre 9 : l'expérimentateur se donne *a priori* l'information selon laquelle le système diffractant est constitué de lignes de résine photosensible disposées sur un substrat de silicium. Il connaît les indices des matériaux en jeu, la période du réseau et la forme que possède le créneau. Cette forme étant fonction de paramètres géométriques, le problème inverse consistera à exploiter la signature acquise pour déterminer ces paramètres.

Sur l'exemple du schéma 3.6, ces paramètres sont la largeur du créneau de résine (CD), sa hauteur H, et l'arrondi de tête r.

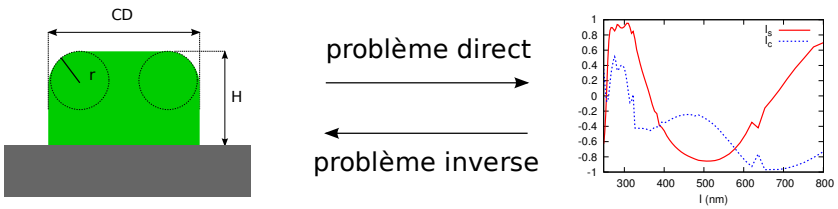


Fig. 3.6: Problèmes direct et inverse en scatterométrie

CHAPITRE 4

Le problème inverse en scatterométrie

Formellement, si l'on défini le problème direct comme le calcul d'une signature scatterométrique **s** à partir d'un jeu de paramètres **p**, pour un modèle $\mathcal{M}$ de créneau diffractant donné :

$$\mathbf{s} = f_{\mathcal{M}}(\mathbf{p})$$

alors le problème inverse consiste à trouver **p** connaissant la mesure **s**.

Ce que l'on formalise par le vecteur signature **s** est, en pratique, un vecteur de taille $N_{\mathbf{s}} = 2N_\lambda$ (N_λ étant le nombre de longueurs d'onde du spectre de la signature) de la forme :

$$\mathbf{s} = (D^1_{\lambda_1}, D^1_{\lambda_2}, ..., D^1_{\lambda_{N_\lambda}}, D^2_{\lambda_1}, D^2_{\lambda_2}, ..., D^2_{\lambda_{N_\lambda}})$$

avec (D^1, D^2) un couple de signaux ellipsométriques (cf. 2.3).

4.1 Distances entre signatures

Ce problème est en fait un problème de minimisation. Ce que l'on cherche est le jeu de paramètres **p** qui engendre une signature **s** la plus semblable possible à la **signature acquise $\tilde{\mathbf{s}}$**, c'est-à-dire qui minimise la distance :

$$\min_{\mathbf{p}} d[f_{\mathcal{M}}(\mathbf{p}), \tilde{\mathbf{s}}]$$

Mais comment choisir cette fonction distance d ?

De manière courante, les techniques d'optimisation se basent sur la méthode du χ^2. Cette méthode consiste à évaluer la distance entre deux signatures ($\mathbf{s} = f_{\mathcal{M}}(\mathbf{p})$ et $\tilde{\mathbf{s}}$) en utilisant la norme euclidienne (ou norme 2) :

$$\chi^2 = \sum_{i=1}^{N_s} [s_i - \tilde{s}_i]^2$$

Cependant, sans le savoir le plus souvent, l'expérimentateur qui utilise cette méthode fait deux hypothèses[1]. Si l'on définit le résidu normalisé d'une mesure comme l'erreur commise lors de l'approximation d'une signature expérimentale par une signature simulée :

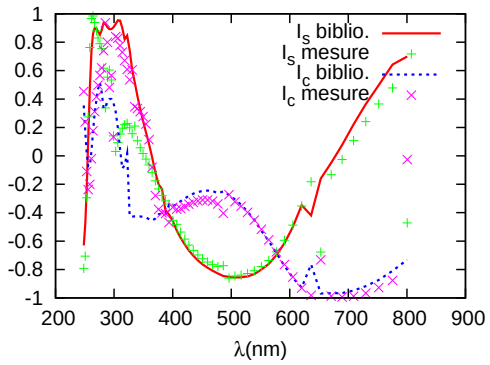
$$f_i = \frac{s_i - \tilde{s}_i}{\sigma_i} \tag{4.1}$$

où $\sigma_i^2 = Var(s_i - \tilde{s}_i)$, alors ces hypothèses sont :

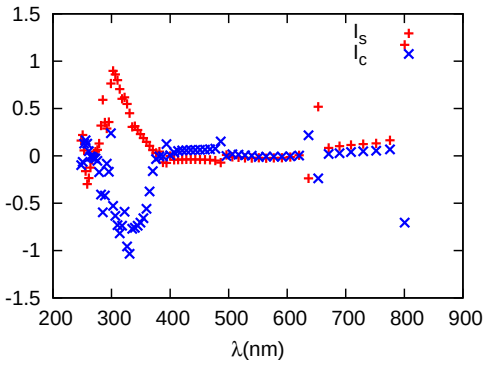
1. les f_i suivent une loi gaussienne ;
2. les f_i sont indépendants (non-corrélés)

Or, bien que l'on puisse considérer que le bruit, ou l'erreur produite lors de la mesure de la signature, est de nature gaussienne, il en est autrement pour l'erreur induite par le calcul électromagnétique (MMFE), par la modélisation géométrique ou par l'utilisation de telle ou telle méthode d'optimisation (cf. section 4.2). Pour ces dernières sources d'incertitudes, il est très délicat de modéliser la statistique et elle n'est probablement pas gaussienne ; la première hypothèse est donc injustifiée.

D'autre part, comme le montre le graphique 4.1, les f_i ne sauraient être indépendants les uns des autres ; des résidus associés à des longueurs d'ondes proches sont eux même proches.



(a) Régression : les points correspondent aux mesures et les lignes aux signaux régressés.



(b) Résidus en valeur relative

Fig. 4.1: On a représenté ici la régression (4.1(a)) et les résidus associés (4.1(b)) pour la recherche dans une bibliothèque d'une signature (acquise) correspondant à un simple créneau de résine (largeur de 150nm, hauteur de 370nm environ) sur silicium. On observe là une relation de continuité entre les résidus.

En résumé, la méthode usuelle du χ^2 gagnerait à être remplacée par une solution utilisant un estimateur plus **robuste** (cf. [43]). Des recherches dans ce domaine sont notamment poursuivies au DOPT [2]. Pour les travaux de cette thèse, nous avons choisi d'utiliser la norme 1, c'est-à-dire la fonction distance suivante :

$$d[\mathbf{s}, \tilde{\mathbf{s}}] = \sum_{i=1}^{n} |s_i - \tilde{s}_i|$$

Cette technique est réputée plus robuste [18] notamment pour approcher des valeurs dont on ne connaît pas la statistique associée. Un exemple visuel de la validité de ce choix est donné au chapitre

[1] Plus de détails et une réflexion plus approfondie sur ce problème pourront être trouvés dans la thèse de Quintanilha [80].

[2] Département d'optique du LETI, au CEA.

7 où l'on observe notamment que la norme 2 introduit un biais dans la résolution du problème inverse lorsqu'on utilise la méthode des bibliothèques (cf. 4.2).

4.2 Méthodes d'optimisation

Le problème qui nous concerne est donc de minimiser la fonction coût suivante :

$$\min_{\mathbf{s}} d[\mathbf{s}, \tilde{\mathbf{s}}] \equiv \min_{\mathbf{s}=(s_1,s_2,\ldots,s_n)} \sum_{i=1}^{n} |s_i - \tilde{s}_i|$$

Une illustration de la topologie d'une telle fonction est donnée sur la figure 4.2 : on a représenté par un dégradé de couleurs les valeurs de la distance d entre :

- une signature scatterométrique acquise $\tilde{\mathbf{s}}$: il s'agit du même créneau de résine que la figure 4.1 (dimensions approximatives $CD = 150nm, H = 370nm$ posé sur un substrat de silicium).
- un ensemble de signatures calculées par le code MMFE pour des créneaux de résine de forme rectangulaire, de dimensions variables (largeur CD, hauteur H) et constitués de matériaux connus.

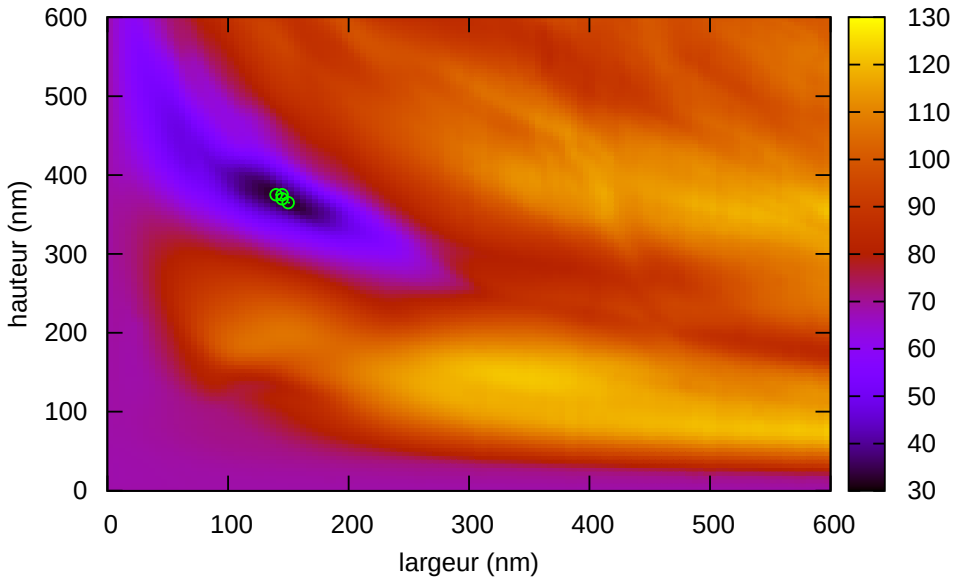


Fig. 4.2: Topographie de la fonction distance entre une signature acquise et un modèle géométrique rectangulaire paramétré par une largeur et une hauteur. Les cercles verts sont localisés dans le minimum global ; ils désignent donc les couples (CD,H) qui décrivent la géométrie qui engendrera la signature la plus proche de $\tilde{\mathbf{s}}$. On peut alors déduire que les paramètres CD et H indiqués correspondent aux dimensions du créneau.

C'est un problème d'optimisation paramétrique **non linéaire** (car les paramètres géométriques **p** du motif ne sont pas reliés linéairement à la signature scatterométrique **s** produite). Ce type de

problème se retrouve dans de nombreux domaines et ceci fait qu'un nombre considérable de techniques de résolutions ont été développées. Nous en présenterons succinctement quelques unes, notamment les plus utilisées pour la scatterométrie (Levenberg-Marquart, méthode des bibliothèques) ou celles faisant l'objet de recherches spécifiques (réseaux de neurones).

On peut classer ces techniques en deux catégories :

1. les techniques d'optimisation *locales* : elles ont souvent l'avantage d'être rapides mais ne promettent de trouver qu'un extremum local, ce qui est problématique si la topologie de la fonction coût est texturée (si elle répète périodiquement un certain motif) ou encore multipolaire (si plusieurs bassins coexistent). On peut citer, parmi ces techniques :
   - **La méthode du simplexe** : Dans un espace de dimension N, un simplexe est l'enveloppe convexe contenant N+1 points. Sur la figure 4.3, le simplexe est un triangle dans un espace de dimension 2 (un plan). Cette méthode d'optimisation repose sur des évaluations de la fonction coût et consiste à déformer un triangle initial en bougeant le plus mauvais point. Sur la figure, où sont représentées des lignes de niveaux, nous avons transformé le triangle initial ABC en substituant A' à A, puis ensuite C' à C, etc.

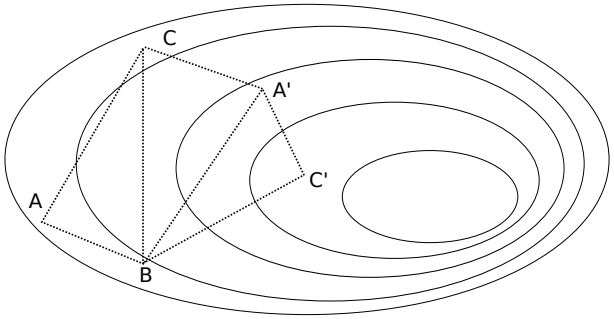
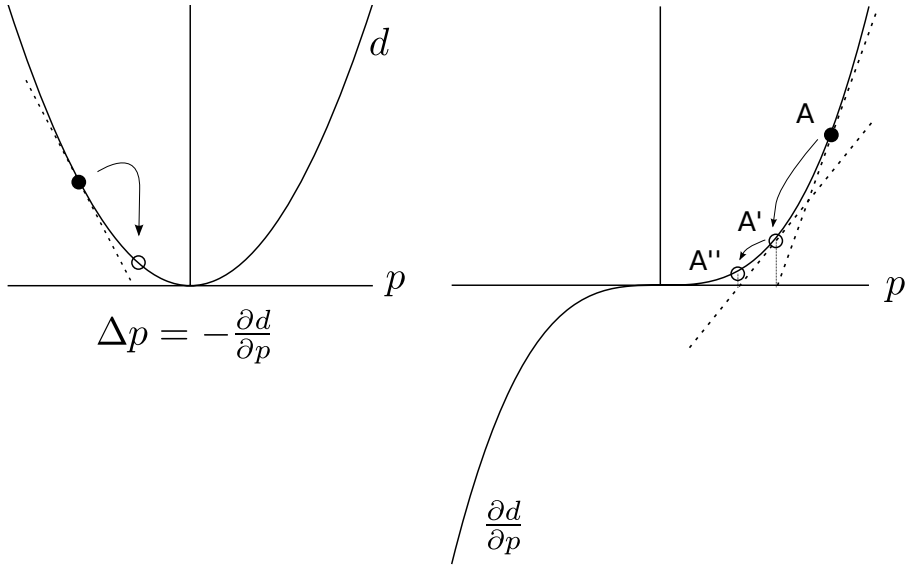


Fig. 4.3: Méthode d'optimisation dite du simplexe, représentée pour deux paramètres.

   - **La méthode Levenberg-Marquart** : cette méthode est une combinaison des méthodes du gradient et de Newton. La méthode du gradient consiste (comme représenté sur la figure 4.4(a)) à trouver le minimum de la fonction coût (d) en déplaçant le vecteur paramètre dans le sens de la pente d'une distance proportionnelle à cette pente. La méthode de Newton se base sur la dérivée de la fonction coût qui est sensée s'annuler à l'optimum. Elle consiste à s'approcher de ce zéro en déplaçant la solution là où la tangente à cette solution coupe l'axe. Sur la figure 4.4(b), A est déplacé en A' puis en A" etc. La méthode du gradient est surtout efficace pour les premières itérations. Ensuite la pente s'amenuise et la solution bouge moins significativement. Au contraire, la méthode de Newton converge significativement surtout aux abords du zéro. La méthode de Levenberg-Marquart est en fait une solution intermédiaire : elle est similaire au gradient en début d'optimisation et devient Newton en fin.
2. les techniques d'optimisation *globales* : l'intérêt ici est d'obtenir à coup sûr la solution optimale dans la plage de paramètres. Malgré cet avantage, quand ces techniques ne sont pas utilisées, c'est à cause du temps de calcul souvent prohibitif. On peut notamment citer, parmi ces méthodes :
   - **Les réseaux de neurones** : C'est une méthode de calcul qui mime le fonctionnement des véritables neurones et synapses du monde du vivant. Cette technique prend appui sur une phase d'apprentissage qui sert à améliorer l'efficacité d'un réseau de fonctions statistiques.



(a) descente du gradient (b) méthode de Newton

Fig. 4.4: Levenberg-Marquart comme méthode d'optimisation intermédiaire entre la méthode du gradient et Newton

Des résultats très encourageants pour la scatterométrie ont été obtenus par Gereige *et al.*([30])

- **Les algorithmes évolutionnaires** (ou génétiques) : Il s'agit ici de reproduire le mécanisme de l'évolution des espèces découvert par Charles Darwin. Une solution, telle un être vivant, doit disparaître si elle est en compétition avec une meilleure (relativement à la fonction de coût) congénère (sélection naturelle). Et elle doit aussi évoluer : on provoque pour cela une mutation (génétique, dans le vivant) à chaque nouvelle génération ; cette mutation pourrait par exemple être une perturbation aléatoire du vecteur **p**. Ce domaine de l'intelligence artificielle à été introduit en 1975 par J.H. Holland [40] et son efficacité (toute relative) pour la scatterométrie a été évaluée par Raymond *et al.* [85].
- **La méthode des bibliothèques** : Potentiellement précise et rapide, c'est cette technique que nous avons choisi pour ces travaux de thèse et que nous allons développer dans la suite.

4.2.1 Méthode des bibliothèques

La méthode des bibliothèques consiste simplement à construire une base de données de signatures scatterométriques. Chaque signature est issue de la simulation (donc calculée avec un code électromagnétique tel que la MMFE, cf. section 3.3), et est indexée par un jeu de paramètres géométriques unique. La résolution du problème inverse consiste alors à comparer la signature acquise à l'ensemble des signatures de la base. Ceci est illustré sur le schéma 4.5. On extrait ainsi le (ou les) jeu(x) de paramètres géométriques qui correspondent le mieux au profil diffractant.

L'avantage de cette méthode d'optimisation est qu'elle est *globale*. En effet, lors de la recherche, la fonction coût sera évaluée sur l'ensemble des signatures stockées et son extremum sera un élément de l'espace entier des paramètres. Un autre avantage est la rapidité avec laquelle on obtient cet extremum : puisqu'il n'y a pas besoin d'exécuter le code électromagnétique (c'est fait au préalable lors de la génération de la base), l'optimisation est une répétition d'opérations simples qui s'exécutent

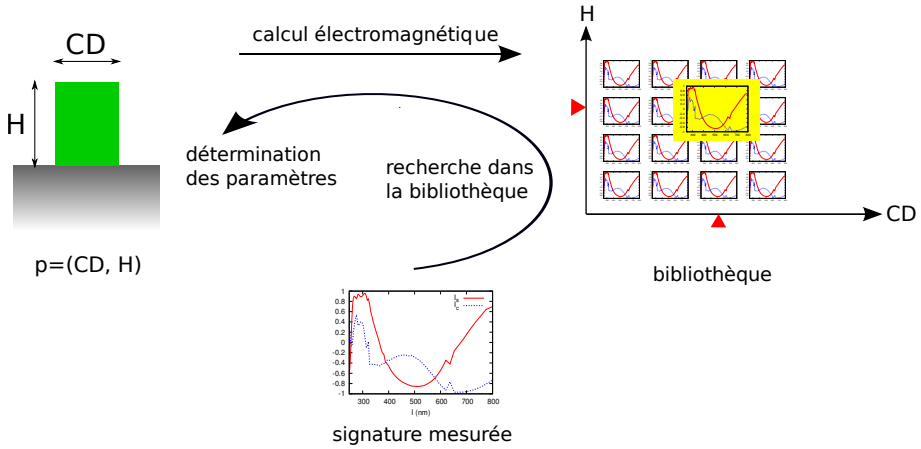


Fig. 4.5: Illustration de la méthode des bibliothèques pour un modèle carré, paramétré par une largeur de créneau CD et une hauteur H

rapidement sur des ordinateurs modernes.

Toutefois, cette méthode exige un temps de calcul considérable : un problème typique (en milieu industriel par exemple) implique le calcul de quelques centaines de milliers (voire millions) de signatures pour 3 ou 4 paramètres géométriques, et chaque signature suppose d'exécuter le code électromagnétique autant de fois qu'il y a de longueurs d'ondes dans le spectre... Le temps de calcul total se chiffre en jours (voire en semaines selon la complexité du modèle) si l'on utilise un seul processeur (La taille de la bibliothèque est alors de l'ordre du Gio [3]).

La bibliothèque logicielle *libScattero* développée durant cette thèse intègre pour cela la capacité de distribuer le calcul de la bibliothèque sur autant de coeurs de microprocesseurs disponibles avec une efficacité parallèle de l'ordre de 100%. Ceci fait que, sur une machine du commerce récente munie de 8 coeurs de calculs par exemple (deux processeurs quadri-coeur), la bibliothèque sera générée 8 fois plus rapidement. Une description détaillée de *libScattero* est fournie en annexe D.

Un autre aspect de cet exigence est qu'il est difficile de modifier ou de raffiner le modèle : chaque paramètre en plus ou chaque changement de plage de variation d'un paramètre suppose le calcul d'une autre bibliothèque.

Cependant, la méthode est utilisée depuis longtemps et elle est éprouvée en milieu industriel pour le contrôle de procédés en ligne (mais pas *in situ* en temps réel). Dans ce cas, une bibliothèque est calculée une fois pour toutes (car le procédé est connu) et les paramètres géométriques extraits servent à ajuster les paramètres de fabrication.

Conclusion : méthode des bibliothèques pour l'optimisation en temps réel. Un élément majeur à prendre en compte pour ces travaux de thèse est la nature **temps réel** du système de métrologie à développer. Ceci signifie que le temps de traitement des données ne doit pas être un facteur limitant du système : Si l'on considère que le rythme d'acquisition des signatures par l'ellipsomètre est régulier, alors le temps de traitement doit être borné et maintenu inférieur à la période de la mesure. Ceci est la *véritable* définition de la contrainte de temps réel, courante en milieu industriel.

Parmi les méthodes de résolution du problème inverse, une seule est réellement compatible avec

[3] Un gibioctet (Gio) équivaut à 2^{30} octets

cette contrainte de temps réel. Il s'agit de la méthode des bibliothèques. D'autres algorithmes itératifs pourraient se voir adaptés au temps réel (c'est-à-dire que l'exécution pourrait être arrêtée au bout d'un temps fixé), mais l'erreur induite par la résolution ne saurait alors être bornée (Levenberg-Marquart, simplexe, etc.)

Les travaux présentés dans ce mémoire consistent donc à étendre la méthode des bibliothèques (elle n'est en principe utilisable que pour la scatterométrie statique) au suivi des procédés dynamiques.

4.3 Scatterométrie en temps réel

Le développement des méthodes de scatterométrie aux suivis de procédés en temps réel n'a pas pour l'instant fait l'objet d'une bibliographie véritablement abondante, d'autant que la notion de temps réel est parfois vue comme "temps de résolution très court", c'est-à-dire quelques secondes.

Dans les travaux de Opsal *et al.* par exemple, l'enjeu est l'amélioration du code de simulation [73] pour permettre une optimisation correcte en utilisant une méthode itérative, cela en quelques secondes seulement. Ces travaux permettent de proposer une solution de métrologie en ligne (c'est-à-dire intégrée à la chaîne de fabrication des semi-conducteurs) [74] ou d'envisager des applications *in situ*. Maynard *et al.* [69] ont la même approche.

Terry *et al.* [42] sont concernés comme nous par le suivi de procédés de gravure *in situ*. L'utilisation d'une signature scatterométrique fortement résolue et d'une méthode d'optimisation itérative (Levenberg-Marquart) semble permettre une reconstruction détaillée de la géométrie du profil diffractant. Peu de choses sont dites cependant sur la façon d'acquérir rapidement de telles signatures et de les traiter en temps réel.

Galarza *et al.* [26] en revanche sont plus explicites, notamment sur l'utilisation d'un filtrage de Kalman pour la reconstruction des paramètres géométriques dans le temps et sur la partition du spectre des longueurs d'onde : une partie sert à affiner le modèle géométrique, l'autre à en déterminer les paramètres.

En conclusion, il semble qu'à ce jour, peu de travaux concernent la scatterométrie temps réel telle que nous l'entendons, c'est à dire une véritable reconstruction en temps réel de paramètres géométriques variant sous l'effet d'un procédé.

CHAPITRE 5

Reconstruction dynamique de paramètres

La scatterométrie dynamique a pour vocation de suivre les variations d'un motif diffractant au cours d'un procédé. Ceci implique que la fréquence d'acquisition des signatures en provenance de l'ellipsomètre soit suffisamment élevée en regard du temps pour lequel le profil varie notablement.

Cependant, une bonne fréquence d'acquisition s'obtient avec une résolution plus faible en longueurs d'onde. Ceci est une limitation matérielle de l'ellipsomètre : une mesure dynamique utilise un nombre restreint de photomultiplicateurs, chacun calibré sur une longueur d'onde particulière. Par exemple, les mesures rapportées dans la partie III sont effectuées avec 16 longueurs d'ondes pour l'ellipsomètre en chambre de gravure et 32 longueurs d'onde pour l'ellipsomètre de la plaque chauffante (fluage). [1]

La faible résolution en longueurs d'onde des signatures nuit à l'acuité de la recherche dans la bibliothèque car on réduit alors le nombre de points de comparaison.

Notre algorithme de reconstruction dynamique de profil a pour objectif de compenser cette faible résolution par une fréquence d'acquisition plus élevée.

Le principe de base est le suivant :

- au premier pas de temps ($t = 0$), c'est-à-dire *avant* le début du procédé de fabrication, le profil diffractant est supposé parfaitement connu. On utilise pour cela n'importe quelle technique de métrologie statique : scatterométrie standard bien sûr, mais aussi AFM (*Atomic Force Microscopy*, cf. 9.1.2) par exemple.
- aux instants suivants :
  1. Une recherche des k signatures les plus proches de la signature acquise est faite dans la bibliothèque.
  2. Un choix est fait parmi les k signatures extraites de la base pour n'en garder qu'une. Ce choix de signature, dont le principe est similaire à une régularisation temporelle de Tikhonov [92], est effectué en prenant en compte non seulement la proximité de cette signature à la signature acquise mais aussi la régularité de la variation des paramètres géométriques du motif.

[1] Ces deux ellipsomètres sont néanmoins capables de fournir une signature mieux définie mais le temps d'acquisition est alors très long, de l'ordre de la minute. On parlera alors d'acquisition *spectroscopique* (car l'objectif est la résolution du spectre en longueurs d'onde) plutôt que *dynamique*.

3. Enfin, une véritable régularisation temporelle (lissage) de Tikhonov est opérée.

Pour faciliter la compréhension de ce qui suit, nous rappelons les notations utilisées dans ce chapitre :

- $\mathbf{p}^t$ est un vecteur de taille $N_\mathbf{p}$ contenant les paramètres géométriques du motif diffractant à l'instant t. Par exemple, $\mathbf{p}^t = (CD^t, H^t, ...)$ pour un profil de créneau déterminé à chaque instant par sa hauteur H, et sa largeur CD, etc.
- $\mathbf{s}^t = f_\mathcal{M}(\mathbf{p}^t)$ est une signature scatterométrique de la bibliothèque calculée par un code de simulation électromagnétique (dans notre cas, MMFE, cf. 3.3), avec un modèle géométrique $\mathcal{M}$ paramétré par $\mathbf{p}^t$ et un ensemble de matériaux définis par $n(\lambda)$ et $k(\lambda)$, respectivement l'indice de réfraction et le coefficient d'extinction. En pratique, c'est un vecteur de taille $N_\mathbf{s} = 2 \times N_\lambda$ (où N_λ est le nombre de longueurs d'ondes) de la forme :

$$\mathbf{s}^t = (D^{1,t}_{\lambda_1}, D^{1,t}_{\lambda_2}, ..., D^{1,t}_{\lambda_{N_\lambda}}, D^{2,t}_{\lambda_1}, D^{2,t}_{\lambda_2}, ..., D^{2,t}_{\lambda_{N_\lambda}})$$

  avec (D^1, D^2) un couple de signaux ellipsométriques tels que (Ψ, Δ), (I_S, I_C), etc. (cf. chapitre 2).
- $\mathcal{B}$ est l'ensemble des signatures calculées par simulation numérique ; c'est la bibliothèque. Chaque signature est fonction d'un ensemble de matériaux et d'un ensemble de paramètres géométriques.
- $\tilde{\mathbf{s}}^t$ est la signature expérimentale acquise à l'instant t (de taille $N_\mathbf{s}$).

5.1 Recherche des k plus proches voisins

La bibliothèque $\mathcal{B} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{N_\mathcal{B}}\}$ est un ensemble de $N_\mathcal{B}$ signatures, chacune étant un vecteur de dimension $N_\mathbf{s} = 2N_\lambda$.

Le problème de la recherche des k plus proches voisins (nous le noterons k-P.P.V. par la suite) consiste à trouver dans l'ensemble $\mathcal{B}$ les k vecteurs $\mathbf{s}_i$ qui minimisent une distance par rapport à un vecteur requête $\tilde{\mathbf{s}}$ de même dimension :

$$\min_i ||\tilde{\mathbf{s}} - \mathbf{s}_i||$$

À titre d'illustration, la figure 5.1 représente un espace normé de 2 dimensions dans lequel figurent plusieurs points. Dans notre cas il s'agit de l'espace des signatures et, dans cet exemple, chaque signature ne contient qu'une seule longueur d'onde. Les coordonnées des points seront donc I_S et I_C. Les contours k-P.P.V. renferment les points les plus proches voisins de la requête $\tilde{\mathbf{s}}$ (signature acquise) en considérant la norme choisie.

La norme choisie est la norme 1 (norme de Manhattan), comme cela est justifié au chapitre 4 :

$$\min_{\mathbf{s}\in\mathcal{B}} ||\tilde{\mathbf{s}}^t - \mathbf{s}||_1 = \min_{\mathbf{s}\in\mathcal{B}} \sum_{i=1}^{N_\mathbf{s}} |\tilde{s}^t_i - s_i| \tag{5.1}$$

Cette recherche de k-P.P.V. sera effectuée à chaque instant t en utilisant éventuellement un processeur graphique comme accélérateur. C'est l'objet du chapitre 6.

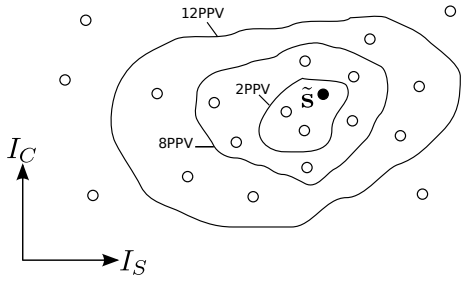


Fig. 5.1: Recherche des k-P.P.V.

5.2 Choix d'une signature parmi les k-P.P.V.

5.2.1 Procédé

Si l'on retire une seule solution $\mathbf{s}$ de $\mathcal{B}$ à chaque pas de temps, la variation des paramètres est directement dépendante de l'erreur de la mesure. Cette erreur engendre une signature déformée et, à cause du peu de longueurs d'ondes acquises, mène à une solution inadéquate : minimum local de la bibliothèque dans lequel les paramètres géométriques sont erronés, variation erratique des valeurs de ces paramètres, etc.

Au contraire, si l'on extrait à chaque instant **plusieurs** plus proches voisins de la bibliothèque, on a le choix entre plusieurs jeux signatures-paramètres, et on peut valoriser une part de régularité dans les variations temporelles des paramètres géométriques.

Pour cela, on calcule à chaque pas de temps t la valeur $J(\mathbf{p}_i^t)$ $(1 \leq i \leq k)$ pour chacun des k vecteurs $\mathbf{p}_i^t$ issu de la recherche dans $\mathcal{B}$ et l'on choisira le candidat i (parmi les k) qui minimise la grandeur $(\forall t \in \mathbb{N}^*)$:

$$J(\mathbf{p}_i^t) = ||\tilde{\mathbf{s}}^t - \mathbf{s}_i^t|| + \Psi(\mathbf{p}^0, ..., \mathbf{p}^{t-1}, \mathbf{p}_i^t, t) \tag{5.2}$$

La fonction Ψ est la **fonctionnelle régularisante**. Elle a pour rôle de favoriser la régularité de la suite des paramètres géométriques $(\mathbf{p}^t)_{t\geq 0}$. Cette fonctionnelle prend en compte les paramètres passés $(\mathbf{p}^0, ..., \mathbf{p}^{t-1})$ pour influer sur le choix de i dans le problème de minimisation 5.2.

On voit ici pourquoi la détermination précise de $\mathbf{p}^0$ est cruciale : c'est ce vecteur qui guidera par régularité le choix de $\mathbf{p}^1$ et des suivants.

Nous avons choisi ici une fonctionnelle régularisante simple :

$$\Psi(\mathbf{p}, t) = \left| \tilde{\beta_1} \cdot \frac{\partial \mathbf{p}}{\partial t} \right|^2 \tag{5.3}$$

qui devient, une fois discrétisée :

$$\Psi(\mathbf{p}^0, ..., \mathbf{p}^{t-1}, \mathbf{p}^t, t) = \left| \tilde{\beta_1} \cdot \frac{\mathbf{p}^t - \mathbf{p}^{t-1}}{\Delta t} \right|^2 \tag{5.4}$$

Le vecteur $\tilde{\beta_1}$ est analogue au paramètre de régularisation de Tikhonov (cf. section 5.3). La détermination de ses coefficients, aussi nombreux que les paramètres géométriques, constitue un élément de la recherche à poursuivre. Il n'est pas possible de tous les combiner en un seul scalaire

β car les coefficients n'ont pas nécessairement le même ordre de grandeur (on pourrait néanmoins les normaliser), et surtout on doit pouvoir choisir la *force* avec laquelle on doit contraindre chaque paramètre géométrique à être régulier.

Cependant, notre choix pour cette fonctionnelle a été guidé par ces considérations :

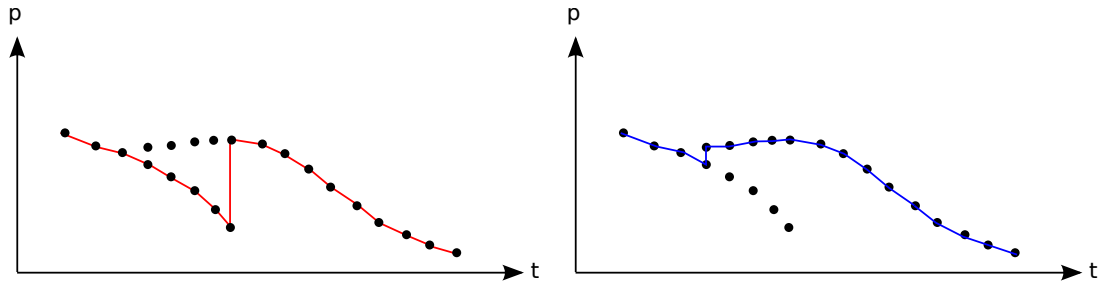
- En premier lieu, nous cherchions à pénaliser des variations brutales de géométrie (elles sont improbables dans un procédé bien maîtrisé) ; la dérivée première de $\mathbf{p}$ par rapport au temps en est significative.
- Nous ne sommes pas confrontés ici à des fréquences d'oscillations parasites ; inutile donc d'inclure la dérivée seconde (cf. [92])
- Un seul terme dans Ψ implique un seul vecteur $\tilde{\beta_1}$ à déterminer, ce qui est bien suffisant en terme de difficulté.

5.2.2 Remarque sur le choix de $\mathbf{p}^t$ en deux étapes

À première vue, la résolution du problème du choix du meilleur vecteur paramètre $\mathbf{p}$ en deux étapes (recherche des candidats puis sélection par la régularité) peut sembler être une source d'erreurs. En effet il n'est pas sûr que le vecteur paramètre de la bibliothèque entière qui minimise le mieux $J(\mathbf{p}^t)$ fasse partie de la présélection 5.1.

Cette manière de faire peut se justifier cependant :

1. On pourrait, lors de la recherche des k-P.P.V., inclure le terme de la contrainte de régularité. Cela aurait certes l'avantage de réduire la recherche à un seul élément, mais le choix à l'instant t serait biaisé car dépendant des instants précédents. Cela pourrait conduire à une variation de $\mathbf{p}$ suivant correctement l'évolution d'un minimum local au cours du temps, mais fausse. Le phénomène de *rattrapage* qui se produit lorsque les k-P.P.V. dépeuplent un mauvais minimum local (à cause d'une distance aux signatures trop élevée) et viennent peupler finalement le bon minimum local arriverait plus tard, le terme de régularité freinant cette correction automatique. Les graphes 5.2.2 illustrent ce principe.
2. La deuxième raison est liée à l'utilisation des processeurs graphiques. La quantité de mémoire sur la carte graphique étant limitée, nous avons choisi de la consacrer essentiellement à la partie *signatures* de la bibliothèque ; les paramètres géométriques n'y sont pas stockés. Il serait possible néanmoins de stocker la totalité de la bibliothèque en ajoutant les jeux de paramètres géométriques, mais cela causerait alors des étapes de calcul supplémentaires sur ces derniers et l'efficacité observée serait moindre.



(a) Suivi d'un minimum local erroné. Correction tardive.

(b) Correction immédiate car la recherche des k-P.P.V. est toujours globale.

Fig. 5.2: Correction de la variation de $\mathbf{p}$ (ici de dimension 1) au cours du temps

5.3 Régularisation de Tikhonov

À l'issue de l'étape précédente, nous avons obtenu une suite de vecteurs paramètres $(\mathbf{p}^t)$ indexés par l'instant t. Ces vecteurs sont tous issus de l'espace **discret** formé par la bibliothèque $\mathcal{B}$ et la suite ne forme donc pas de variations régulières des paramètres. Ainsi, la dernière étape de notre procédé est une étape de *lissage*, c'est-à-dire une étape de détermination des paramètres $\mathbf{p}$ dans un espace continu.

Afin de produire une variation régulière de chacun des paramètres géométriques pris indépendamment, plusieurs méthodes existent. Parmi celles-ci nous avons choisi d'utiliser la **régularisation de Tikhonov**[2] [92].

Cette méthode revient à résoudre le problème de minimisation suivant pour chacun de ces paramètres p (p est donc l'une des coordonnées de $\mathbf{p}$) :

$$\underset{\{p\}}{min}\left(\|\hat{p}-p\|_2^2+\beta_2\Omega(p,t)\right) \tag{5.5}$$

Où $\hat{p}$ est la valeur du paramètre choisi (à un instant t quelconque) dans l'étape précédente (cette valeur fait donc partie de la bibliothèque), Ω une nouvelle fonctionnelle régularisante et β_2 le paramètre de régularisation.

Ce problème consiste donc à déterminer les valeurs que prennent les termes d'une suite $(p^t)_{t\geq 0}$ solution du problème de minimisation. Cela n'est pas trivial si la fonctionnelle régularisante Ω prend en compte les valeurs de p à plusieurs instants ; et d'autre part il est évident que plus le nombre d'éléments de la suite est grand, plus le temps de détermination des p^t est important. Ainsi, pour maintenir la compatibilité avec la contrainte de temps réel qui impose un temps de calcul borné, nous avons choisi de réaliser cette régularisation dans une fenêtre de temps glissante et de largeur constante. Nous nommerons $\mathcal{F}\subset\mathbb{N}$ une telle fenêtre (schématisée sur la figure 5.3) et $N_\mathcal{F}=card(\mathcal{F})$ sa taille constante (en nombre de pas de temps).

Si la fonctionnelle régularisante $\Omega(\mathbf{p})$, une fois discrétisée, prend en compte, à l'instant t, des valeurs de $\mathbf{p}$ à des instants antérieurs ($\mathbf{p}^{t-1}$, $\mathbf{p}^{t-2}$, etc) ou postérieurs, alors la première régularisation sera retardée en conséquence. Si par exemple $\Omega(\mathbf{p},t)=f(\mathbf{p}^{t-1},\mathbf{p}^t,\mathbf{p}^{t+1})$ alors, la première fenêtre à être régularisée le sera à l'instant $t=N_\mathcal{F}+1$:

$$\mathcal{F}_1=[1,\ldots,N_\mathcal{F}]$$

et, de manière générale, à chaque instant $T\geq N_\mathcal{F}+1$ auquel une signature est acquise, on effectuera la régularisation dans la fenêtre :

$$\mathcal{F}_{T-N_\mathcal{F}}=[T-N_\mathcal{F},\ldots,T-1]$$

Le problème de minimisation devient donc, à l'intérieur d'une fenêtre glissante $\mathcal{F}$:

$$\underset{\{p^t\}_{t\in\mathcal{F}}}{min}\left(\left\|\hat{p}^t-p^t\right\|^2+\beta_2\Omega(p)\right) \tag{5.6}$$

Fonctionnelle régularisante. Le choix de $\Omega(p,t)$, tout comme celui de Ψ à la section précédente, est un point à développer. Pour notre application, nous en avons choisi une très élémentaire, sensiblement équivalente à Ψ :

$$\Omega(p)=\int_\mathcal{F}\left(\frac{\partial p}{\partial t}\right)^2 dt$$

[2] L'efficacité de cette solution s'étant avérée tout à fait satisfaisante, nous n'avons pas remis ce choix en cause par la suite.

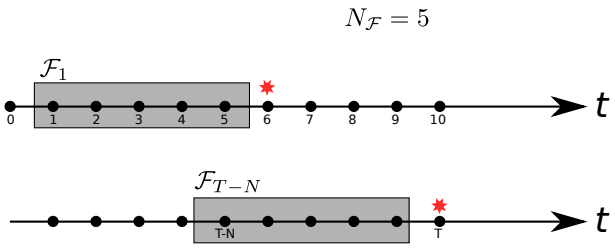


Fig. 5.3: Fenêtre de régularisation glissante au cours du temps. Les étoiles représentent le dernier vecteur **p** issu de l'étape de pré-régularisation 5.2.

Cette fonctionnelle a été choisie pour les mêmes raisons que Ψ. Elle a en plus l'avantage de mener à un système linéaire tridiagonal donc soluble très rapidement par une méthode de décomposition L-U telle Crout ou Doolitle [79].

En effet, Ω se discrétise facilement dans une fenêtre de temps $\mathcal{F}_{T-N_{\mathcal{F}}}$:

$$\Omega^T = \sum_{i \in \mathcal{F}_{T-N_{\mathcal{F}}}} \left(\frac{p^{i-1} - p^i}{\Delta t} \right)^2$$

Le problème à résoudre est donc finalement de minimiser $K(p^t)$ $(\forall t \in \mathcal{F}_{T-N_{\mathcal{F}}})$:

$$\begin{aligned} K(p^t) &= (p^t - \hat{p}^t)^2 + \beta_2 \sum_{i \in \mathcal{F}_{T-N_{\mathcal{F}}}} \left(\frac{p^{i-1} - p^i}{\Delta t} \right)^2 \\ \frac{\partial K(p^t)}{\partial p^t} &= 2(p^t - \hat{p}^t) + \frac{\beta_2 2 p^t}{\Delta t^2} [(p^t - p^{t+1}) - (p^{t-1} - p^t)] \\ 0 &= p^{t-1}(-\beta) + p^t(1 + 2\beta) + p^{t+1}(-\beta) - \hat{p}^t \end{aligned} \tag{5.7}$$

Dans l'égalité 5.7 on a substitué $\frac{\beta_2}{\Delta t^2}$ par β pour simplifier l'écriture. Ce qui revient au système tridiagonal suivant :

$$\begin{bmatrix} 1+2\beta & -\beta & & & \\ -\beta & 1+2\beta & -\beta & & \\ & \ddots & \ddots & \ddots & \\ & & -\beta & 1+2\beta & -\beta \\ & & & -\beta & 1+2\beta \end{bmatrix} \begin{bmatrix} p^{T-N_{\mathcal{F}}} \\ \vdots \\ \vdots \\ p^{T-1} \end{bmatrix} = \begin{bmatrix} \hat{p}^{T-N_{\mathcal{F}}} + \beta p^{T-N_{\mathcal{F}}-1} \\ \hat{p}^{T-N_{\mathcal{F}}+1} \\ \vdots \\ \hat{p}^{T-2} \\ \hat{p}^{T-1} + \beta p^T \end{bmatrix}$$

On remarque que les valeurs de $p^{T-N_{\mathcal{F}}-1}$ et p^T sont impliquées dans la régularisation des p^t $(t \in \mathcal{F}_{T-N_{\mathcal{F}}})$ et ne sont pas changées. $p^{T-N\mathcal{F}-1}$ est le dernier p^t à être sorti de la fenêtre de régularisation et p^T est en réalité $\hat{p}^T$, la dernière valeur issue de la pré-régularisation. On comprend aussi pourquoi la première acquisition est importante : $\hat{p}^0$ ne sera jamais changé car jamais régularisé, ni lors du choix du vecteur paramètre $\mathbf{p}^t$ parmi k, ni lors de cette régularisation temporelle de Tikhonov. $\hat{p}^0$ constituera véritablement un pilier sur lequel se baseront les deux régularisations pour ne pas plonger dans de mauvais minimums locaux dés le début du suivi de procédé.

Optimisation du paramètre β. Ce paramètre est ici assez facile à déterminer. On trace pour cela la courbe en L correspondant au problème 5.6 : il s'agit de l'ensemble des points ayant pour coordonnées le membre de gauche et le membre de droite de l'équation 5.6 :

$$P_\beta = \left(\|\hat{p} - p\|_2^2 , \Omega(p,t) \right)$$

Cet ensemble est tracé (au besoin en échelle logarithmique) pour des valeurs différentes de β.

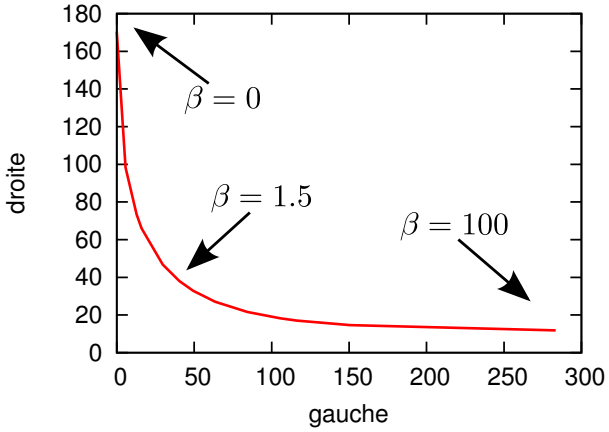


Fig. 5.4: Courbe en L correspondant au lissage du paramètre CD dans l'exemple simulé 7.

Pour notre fonctionnelle, cette courbe a une forme en L (voir la figure 5.4) et le paramètre β qui minimise le problème se situe dans le creux[3]. Ce choix est certes optimal du point de vue numérique (il minimise les résidus) mais ne correspond pas forcément au lissage voulu. Il s'agit à nouveau d'une question d'information *a prori* : si l'utilisateur sait que le procédé à suivre est très régulier, alors il forcera le lissage. Au contraire, s'il souhaite observer des variations précises sur de courtes périodes, alors il prendra garde à fixer un β le moins grand possible.

5.4 Conclusion : comparaison avec les filtres de Kalman

Notre procédé de reconstruction s'apparente à ce qui pourrait par exemple servir pour le suivi de cible (notre vecteur $\mathbf{p}$ à en effet pour ambition de suivre un minimum mouvant au cours du temps).

Pour résoudre ce problème, il est courant d'utiliser les filtres de Kalman [4]. Ce sont des estimateurs récursifs : dans une version linéaire, un filtre de Kalman estime l'état courant ($\mathbf{p}^t$) en fonction de l'état précédent ($\mathbf{p}^{t-1}$) et de la mesure actuelle ($\mathbf{s}^t$). Et l'état du filtre est caractérisé par l'état courant et la matrice de covariance qui estime sa précision.

L'estimation de $\mathbf{p}^t$ se fait en deux étapes :

- une étape de prédiction dans laquelle $\mathbf{p}^t$ est prédit à l'aide de $\mathbf{p}^{t-1}$ et du modèle d'évolution qui doit être linéaire (les états $t-1$ et t sont reliés linéairement)
- une étape de mise à jour dans laquelle la nouvelle mesure $\mathbf{s}^t$ sert à corriger $\mathbf{p}^t$ en améliorant sa précision. Là aussi, le modèle d'observation qui relie mesure et état doit être linéaire.

[3] Tout est ensuite dans la manière de déterminer les coordonnées de ce creux [45] ; on peut par exemple choisir le paramètre β qui minimise le produit des deux membres.

[4] Une des premières implémentation de ce filtre a notamment été faite dans les ordinateurs de navigation du programme Apollo.

Bien entendu, dans notre cas (scatterométrie), la mesure **s** et l'état **p** ne sont pas reliés linéairement. On peut considérer néanmoins que les fonctions qui les relient sont différentiables. On peut alors les linéariser localement (autour de la position courante) et établir à chaque instant la matrice jacobienne J contenant les éléments :

$$J_{i,j} = \frac{\partial p_i}{\partial s_j}$$

où $0 \leq i \leq N_{\mathbf{p}}$ et $0 \leq j \leq N_{\mathbf{s}}$

Ces filtres, adaptés aux cas non linéaires mais différentiables, s'appellent filtres de Kalman étendus.

Nous n'avons pas implémenté ces filtres durant cette thèse[5] ; il est donc impossible d'effectuer de comparaison chiffrée ou visuelle. Néanmoins :

- Le calcul de la jacobienne a chaque pas de temps nous semble compatible avec une application temps réel. En utilisant une bibliothèque assez dense et judicieusement structurée, le calcul des multiples dérivées de la jacobienne peut se faire rapidement (éventuellement au préalable).
- Dans leurs articles [87] et [88], Schmitt *et al.* comparent une méthode de régularisation temporelle semblable à la notre (à l'utilisation d'une bibliothèque près) élaborée pour l'imagerie médicale et les filtres de Kalman étendus : l'efficacité des deux méthodes pour la tomographie est jugée similaire mais la première requiert moins de paramètres à ajuster.
- La convergence des filtres de Kalman est locale, alors que notre procédé permet, on l'a vu, une convergence globale.

La méthode de reconstruction de profil présentée ici constitue donc une alternative aux filtres de Kalman. Elle constitue une amélioration de la très répandue méthode des bibliothèques et, implémentée en utilisant les processeurs graphiques, elle représente une solution viable, de convergence globale, pour le suivi temps réel de signatures scatterométriques. Une démonstration de ce procédé de reconstruction sur une expérience artificielle est exposée au chapitre 7.

[5] On pourra néanmoins consulter l'article de Galarza *et al.* [26] traitant de suivi temps réel en ellipsométrie spectroscopique et notamment de l'établissement du modèle géométrique : la méthode des filtres de Kalman étendus y est comparée.

CHAPITRE 6

Processeurs graphiques et k plus proches voisins

L'algorithme de reconstruction dynamique de paramètres géométriques exposé dans le chapitre précédent (chap. 5) s'appuie, à chaque mesure, sur une recherche dans une bibliothèque des k signatures les plus proches de la signature relevée par l'ellipsomètre.

Lorsqu'on modélise géométriquement un motif de résine avec beaucoup de paramètres (rarement plus de 10, toutefois) ou lorsque l'on souhaite une densité importante de la bibliothèque, alors il est nécessaire de recourir à une quantité de données considérable. Pour le suivi en ligne, l'industrie de la microélectronique utilise des bibliothèques occupant de l'ordre du demi Gio ou du Gio en mémoire.

Au delà du fait que la génération de cette bibliothèque en un temps raisonnable demande de gros moyens de calculs (calcul multiprocesseurs, clusters, etc.), c'est le temps de la recherche dans une telle bibliothèque qui peut poser problème. En effet, si l'on peut considérer qu'un système ellipsométrique fourni au mieux quinze signatures par seconde, il faut que la recherche dans la bibliothèque occupe en temps, moins d'un quinzième de seconde. Or, le temps de recherche des k signatures les plus proches dans une bibliothèque de 500 Mio (pour des conditions habituelles en scatterométrie : 4,2 millions de signatures de 16 longueurs d'ondes) en utilisant un processeur standard d'ordinateur (que nous nommerons dans la suite CPU, *Central Processing Unit* ; il s'agit dans notre cas d'un processeur Intel Xeon 3.0GHz) est de l'ordre d'une demi seconde en utilisant l'algorithme le plus efficace (et ce, quelle que soit la valeur prise pour le nombre k, cf. tab. 6.1).

k	2	4	8	12
temps de résolution (s)	0.48	0.51	0.65	0.65

Tab. 6.1: Temps de recherche de k signatures plus proches voisines dans une bibliothèque de 512 Mio.

Certes, il est probablement possible d'obtenir, maintenant ou dans le futur, de meilleurs résultats en utilisant des processeurs plus rapides et des mémoires au temps d'accès plus faible. Mais ces chiffres démontrent qu'un processeur de PC standard n'est pas adapté à ce que l'on désire développer, à savoir un système de suivi temps réel intégré à un ellipsomètre standard. Plus généralement, cela démontre que l'architecture *scalaire* d'un tel processeur n'est pas suffisamment efficace et qu'un changement de type de processeur (et donc un changement dans la manière de parcourir la bibliothèque pour trouver les k-PPV) pourrait s'avérer positif.

Au début de ces travaux de thèse en 2005, un petit nombre de chercheurs remarquèrent la nature particulière des processeurs graphiques présents sur les PC et utilisés alors essentiellement pour accroître le réalisme des jeux vidéos. Ils proposèrent d'utiliser ces processeurs nouvellement programmables pour le calcul scientifique. Nous avons alors choisi de nous inscrire dans ce mouvement novateur et d'évaluer ces processeurs pour notre application de scatterométrie temps réel.

6.1 Origine et architecture des processeurs graphiques

6.1.1 Architecture vectorielle et stream processing

Ainsi, nous avons choisi d'évaluer l'architecture *vectorielle* des processeurs graphiques pour notre application car l'architecture *scalaire* des processeurs standards de PC nous paraissait être en défaut. Ces termes correspondent à la manière dont sont traitées les données par le processeur :

- Par une architecture de type **scalaire**, les données sont traitées successivement par l'unité logique (le processeur ou, dans le cas de processeurs multi-coeurs, un des coeurs). On parle aussi d'architecture SISD, de l'anglais *Single Instruction, Single Data* : une seule donnée traitée à la fois par une instruction.
- Par une architecture de type **vectorielle**, c'est un groupe de données qui est traité simultanément. On parle ici de SIMD : *Single Instruction, Multiple Data* : une seule instruction, plusieurs données.

Le type d'architecture scalaire est très couramment utilisé ; tous les processeurs d'ordinateurs personnels (de type x86 fabriqués par Intel et AMD, ou de type PowerPC fabriqués par IBM et Motorola, par exemple) en font partie[1]. Ils sont aussi utilisés aujourd'hui pour la construction de la plupart des super-calculateurs parallèles : dans la liste de juin 2008 des 500 machines les plus puissantes du monde, 498 sont composées de processeurs scalaires.[2].

Ainsi, pour le calcul scientifique du moins, l'architecture vectorielle semble marginalisée (d'autant plus que les 2 calculateurs vectoriels sont loin du haut du classement). Cela n'a pas toujours été le cas : dans les années 70 et jusqu'à la fin des années 80, ce type d'architecture dominait le marché notamment avec les machines Cray. Les processeurs étaient alors très spécialisés et leurs coût de développement devenaient tels que les constructeurs sont passés au *massivement parallèle*, c'est-à-dire à la juxtaposition de processeurs scalaires.

Les processeurs graphiques sont, comme on le verra de manière plus détaillée dans la suite, de nature vectorielle. Leur mode de fonctionnement est même désigné en anglais sous le terme de *stream processing*, ce que l'on pourrait traduire par "traitement de données par flux". Sans entrer dans les détails (une revue théorique peut être trouvée en [91]), de tels systèmes sont caractérisés par de multiples sous-unités de calculs (*modules*) reliées entre elles (par des *canaux*) et sont destinés à traiter de grosses quantités de données, éventuellement en temps réel. Un point important à remarquer est que, dans notre cas (utilisation du GPU pour la résolution du problème inverse), le flux désigne la bibliothèque en mémoire (cette grande quantité de données est considérée *en théorie* comme infinie) et non pas le flux des signatures acquises.

[1] Il est cependant remarquable que ces processeurs, pourtant scalaires par nature, intègrent dans leurs versions les plus récentes une unité logique vectorielle, la plupart du temps destinée à accélérer les applications multimédia : il s'agit par exemple de la technologie AltiVec intégrée aux PowerPC ou SSE (*Streaming SIMD Extensions*) des processeurs Intel x86.

[2] On pourra consulter, à ce sujet, le site internet `www.top500.org`.

6.1.2 Historique des processeurs graphiques

Que ce soit pour des applications professionnelles (C.A.O.[3], traitement d'image, montage vidéo, etc.) ou ludiques (consoles de jeux, ordinateurs personnels), il a rapidement été nécessaire, pour accroître l'efficacité ou le réalisme de l'affichage, de faire appel à des circuits spécialisés. Cela a permis, d'une part, de décharger le processeur central de certains calculs et, d'autre part, de procéder de manière indépendante à l'amélioration d'un nouveau type d'architecture de processeur.

Si les premiers microprocesseurs dédiés n'avaient souvent qu'un rôle de déplacement d'images en deux dimensions (mouvement d'un personnage dans un jeu...), l'avènement de la troisième dimension et l'amélioration du réalisme des rendus ont conduit à donner à ces circuits un double rôle : gestion de mémoire extrêmement rapide pour la 2D (affichage fluide d'un bureau, d'une image ou d'un film) et calculateur 3D pour effectuer le plus rapidement possible les opérations de géométrie dans l'espace (translation, rotation), les projections de lumière, etc.

Ceci a fait que rapidement, le simple ordinateur personnel s'est trouvé muni, en plus de son processeur principal, d'un coprocesseur doté d'un espace mémoire dédié. Ce type de processeur deviendra rapidement programmable (de manière très restreinte au début, toutefois) et potentiellement utilisable pour de nombreuses applications nécessitant l'exécution d'un code peu complexe sur une grande quantité de données.

L'idée d'utiliser les processeurs graphiques pour le calcul scientifique a réellement été expérimentée à partir du moment où sont sorties les versions des grandes bibliothèques graphiques (*OpenGL* et *DirectX*, cf. section 6.2) permettant de programmer les **shaders**, ces multiples petites sous-unités de calcul des GPU responsables du traitement des **textures**, c'est-à-dire des images bi-dimensionnelles stockées dans la mémoire de la carte. Ces *shaders* doivent leur nom à leur fonction originelle : ombrage (gestion de la luminosité) de chacun des points de la texture, les **texels** (de *texture element*, analogue au pixel *picture element*, l'unité élémentaire d'une image numérique).

À l'époque, l'utilisation des GPU souffrait de nombreux écueils : sous-unités de calcul nombreuses mais peu rapides, quantité de mémoire à disposition limitée, calculs peu précis (ne respectant pas notamment la norme IEEE 754 pour les nombres à virgules flottantes), nombreux dysfonctionnements. Ces deux derniers problèmes n'ayant aucune influence visible sur l'écran du joueur, ils ne furent réglés que tardivement.

Dans cette thèse, nous avons utilisé exclusivement des GPU fabriqués par le constructeur taïwanais NVIDIA, et ceci pour plusieurs raisons :

- C'est le premier constructeur à avoir fourni un véritable langage (avec compilateur et bibliothèques) dédié à son matériel permettant l'utilisation détournée de la carte graphique.
- C'est le constructeur qui a le plus communiqué sur son architecture matérielle. Néanmoins, tout est relatif : c'est plus par l'utilisation approfondie des bibliothèques fournies et des mesures de performance qu'a pu se révéler, parmi les spécialistes de la programmation GPU, une partie du mode de fonctionnement.
- Même si aujourd'hui ATI (nouvellement : AMD) commence à vanter la puissance de calcul brute de ses GPU, NVIDIA est encore le seul qui promeut véritablement son architecture pour les applications scientifiques en fournissant de véritables unités de calcul ainsi que les outils logiciels dédiés. Intel, qui à ce jour fait preuve d'une grande volonté à faire sa place sur le marché des GPU pour ordinateurs personnels, se refuse toujours à des applications scientifiques.

Pour ces travaux de thèse nous avons pu tester trois GPU NVIDIA. Ils sont particulièrement représentatifs, dans leur architecture, de la prise en compte progressive par cette marque des utilisa-

[3] Conception assistée par ordinateur

tions non-graphiques. Il s'agit :

- En 2006, d'un processeur de la série 7 : 7900 GTX, 512 Mio de mémoire. Uniquement destiné au graphisme, son utilisation pour le calcul est cependant possible en utilisant la suite logicielle *Cg* destinée à la programmation des *shaders*. Sa mauvaise précision lors des calculs à virgule flottante et ses difficultés à exécuter correctement certaines instructions de boucles et de branchements rendaient alors difficile la mise au point des programmes. Des résultats prometteurs nous ont cependant incité à poursuivre l'effort.
- En 2007, d'un processeur de la série 8 : 8800 GTX, 768 Mio de mémoire. La grande nouveauté de ce GPU est son architecture dite *unifiée* : Auparavant, les sous-unités de calculs dédiées aux textures (shaders) que l'on utilisait étaient distinctes de celles dédiées aux transformations géométriques dans l'espace (*vertex*) ; dorénavant, ces unités sont polyvalentes et s'appellent des processeurs de flux (*stream processors*)[4]. Elles sont au nombre de 128, respectent maintenant le format de nombre à virgule flotante IEEE 754 et se montrent bien plus stables vis à vis des boucles et des branchements. Comme marque de sa volonté à promouvoir l'utilisation scientifique de sa technologie, le constructeur NVIDIA a sorti cette puce en même temps qu'une nouvelle suite logicielle exclusivement destinée au calcul sur GPU, CUDA (cf. section 6.2), et que la gamme de calculateurs Tesla (cartes et unités externes contenant un ou plusieurs GPU, une grosse quantité de mémoire, mais pas de sortie pour un écran d'ordinateur. On assiste là au retour des machines vectorielles, abandonnées depuis les machines Cray.).
  Nous avons choisi de conserver nos programmes *Cg* et, naturellement, l'exécution s'est trouvé largement accélérée.
- En 2008, d'un processeur de la série GT200 : GTX280, 1024 Mio de mémoire. Il s'agit d'une évolution en volume de la génération précédente : plus de transistors (1,4 milliards contre 680 millions) parce que plus de *stream processors* notamment (240 dorénavant).

L'intérêt de multiplier les unités de calcul dans ce type d'architecture est d'augmenter en proportion la puissance disponible. Le tableau 6.2 établit une comparaison entre les GPU utilisés et l'un des plus puissant processeur de PC actuel : Intel Core 2 Extreme Kentsfield QX6800, 4 coeurs, 3GHz. Si l'on considère le nombre d'opérations à virgules flottantes (GFLOPs, *Giga-Floating Point Operations per second*) réalisables par seconde par ces divers processeurs, on observe que la montée en puissance des GPU est de nature quasiment exponentielle.

Modèle	**Commercialisation**	**GFLOPs**	**transistors** (millions)	**shaders**
7900 GTX	début 2006	250	278	24
8800 GTX	fin 2006	518	686	128
GTX 280	mi 2008	933	1 400	240
Intel Core 2 Extreme	2007	52	585	

Tab. 6.2: Comparaison de divers microprocesseurs

La lecture de ce tableau ne doit pas faire penser que l'on a tout intérêt à substituer les processeurs standards (CPU) par des processeurs graphiques. Ces derniers doivent leur puissance essentiellement à la quantité de sous-unités logiques qui travaillent en parallèle et ne sont efficaces que pour certains types de calcul. Ces calculs sont caractérisés par une collection de petits programmes appliqués à une grande quantité de données. Un GPU remplace certes avantageusement le CPU pour le traitement

[4]Dans la suite, on continuera d'utiliser le terme *shader*.

d'image par exemple (il a été conçu pour cela), mais jamais, tel qu'il est conçu aujourd'hui, ne sera le lieu d'exécution d'un système d'exploitation. Cependant, à cause notamment de la difficulté qu'ont les concepteurs de CPU à augmenter la vitesse de fonctionnement de leur système, on assiste aujourd'hui à des développements allant vers une solution hybride : multiplication de coeurs de CPU dans une même puce, conception hétérogène (plusieurs unités logiques de nature différente) pour le processeur Cell (IBM/Sony/Motorola), etc.

Une dernière remarque à propos de l'augmentation de la puissance des cartes graphiques : on assiste aujourd'hui, parallèlement au développement du nombre de shaders, à la mise en parallèle de plusieurs GPU. Pour les jeux, si deux GPU sont présents sur un système, l'affichage des images se fera alternativement (ou plus rarement, ils pourront se partager l'écran en deux) et, pour le calcul sur GPU, la bibliothèque CUDA permet maintenant d'adresser individuellement chacun des sous-systèmes.

6.1.3 Architecture du GPU utilisé

Tous les tests effectués durant la thèse l'ont été sur les trois GPU dont nous disposions. Cependant, l'architecture et les résultats ne seront exposés que pour le dernier d'entre eux, le GPU GTX280 (muni ici de 1 Gio de mémoire). Cette puce est à ce jour la plus élaborée et la plus puissante conçue par NVIDIA.

Au niveau théorique, on l'a vu, on peut considérer un GPU comme une unité de calcul vectorielle et même *de flux*. Au niveau de l'architecture cependant, on doit admettre que ces caractéristiques sont issues d'une construction qui s'approche d'avantage du *massivement parallèle*, c'est-à-dire, assemblant une multitude de sous-unités scalaires.

On peut en effet considérer le GPU GTX280 comme un assemblage de plusieurs niveaux d'unités de calcul et de mémoires. Cette vision des choses est certes partielle (elle masque notamment la partie logique qui coordonne l'action des sous-éléments : ordonnancement des tâches, gestion de la mémoire, etc.) et schématique[5] mais se concentre sur l'essentiel de ce qui nous intéresse dans ces travaux de thèse :

- Comme le montre la figure 6.1, notre GPU est ainsi composé de 10 unités de calculs appelées **TPC**, comme *Thread Processing Cluster* (traduisible par : groupe de calcul des tâches parallèles), ainsi que des espaces mémoires dont une liaison vers la mémoire principale (qui est physiquement sur d'autres puces de la carte graphique).
- Chaque TPC (voir le diagramme 6.2) contient 3 **SM** (*Shader Multiprocessor*), 8 unités dédiées à la gestion des textures (TF : *Texture Filtering unit*) et un espace de mémoire cache.
- Chaque SM contient lui-même 8 unités de calcul élémentaires, les *shaders* munis d'une mémoire commune et de liaisons très rapides.

Ainsi, au niveau global, notre GPU est ainsi constitué de $10 \times 3 \times 8 = 240$ *shaders*. La pertinence du modèle *stream processing* pour décrire son fonctionnement est assuré par le système d'ordonnancement : celui-ci répartit les instructions à exécuter dans chaque SM de manière à masquer d'éventuels temps de latence ; ces unités ne sont en effet pleinement fonctionnelles que si la file des instructions à exécuter est suffisamment pleine pour qu'aucun des *shaders* ne soit inactif.

[5] Donner plus de détails serait d'une part impossible, car le constructeur ne divulgue que peu d'informations, et de toutes façons nuirait à la clarté du propos.

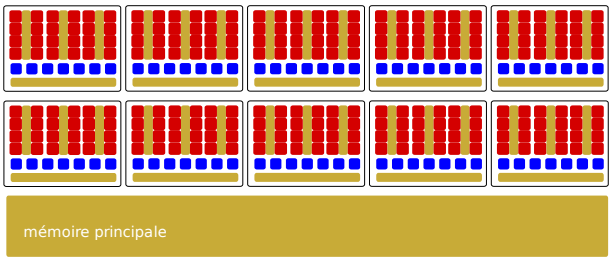


Fig. 6.1: Diagramme de l'architecture globale du GPU GTX280

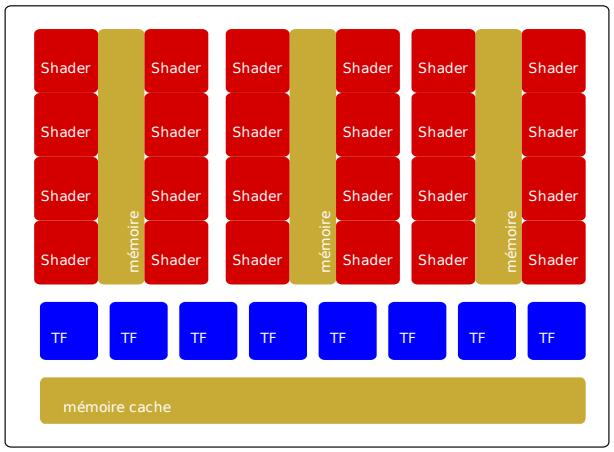


Fig. 6.2: Diagramme d'un TPC

6.1.4 Utilisation scientifique des GPU

Grâce à ces nouveaux GPU devenus programmables, on assiste depuis peu de temps (depuis le début de cette thèse, en pratique) à un retour de l'architecture vectorielle dans le domaine du calcul scientifique : en France, et pour la première fois au monde, sera construit en 2009 au CEA de Bruyères-le-Châtel (dans les salles du Centre de Calcul Recherche et Technologie) le premier super-calculateur utilisant des GPU (48 clusters NVIDIA/Tesla de 4 GPU)[6].

Avant d'avoir acquis la qualité et la crédibilité suffisantes pour être intégrés à un supercalculateur, les GPU ont fait l'objet ces dernières années de nombreux tests et démonstrations. A été évaluée en particulier leur capacité à résoudre certains problèmes de mathématiques appliquées ou de physique de manière efficace. Ce domaine d'activité est souvent dénommé **GPGPU**, de *General Purpose GPU*, c'est-à-dire l'utilisation des GPU dans un but général, et non plus forcément graphique.

Les applications sont devenues aujourd'hui très nombreuses (lire à ce propos [93], [75] et surtout consulter le site internet `www.gpgpu.org`). S'il n'est pas opportun de les citer toutes ici, nous pouvons cependant en dégager quelques-unes particulièrement démonstratives ou ayant un rapport avec nos travaux :

- bases de données numériques : [31], [11]
- vision par ordinateur : suivi de forme (utilisation du problème des k-P.P.V.) : [68], [67]
- imagerie médicale : reconstruction tomographique : [95]

[6] On pourra consulter à ce propos la page : `http://www.genci.fr/spip.php?article23`.

– phénomènes physiques : problème à N corps, mécanique des fluides, équation de la chaleur, etc.

6.2 La programmation des GPU

6.2.1 Spécificités de l'architecture GPU

L'architecture dédiée au graphisme des GPU impose des contraintes et des différences dans la manière de programmer. L'utilisation des outils disponibles au début de nos travaux de thèse (*OpenGL*, *Cg*, cf. 6.2.2) imposent de considérer le calcul proprement dit comme une série d'effets graphiques appliqués sur des textures (données d'entrée). Et la sortie, le rendu final, doit être aussi une image, projetée sur un écran virtuel.

De plus, les contraintes imposées par l'architecture sont les suivantes :

– La mémoire implantée sur la carte est certes très rapide mais elle est limitée et n'est pas extensible pour le moment.
– Si la lecture aléatoire en mémoire (gathering) est possible (on peut utiliser plus d'un texel d'une texture d'entrée pour en calculer un en sortie), l'écriture aléatoire (*scattering*) est en revanche impossible : tous les *texels* (éléments) de la texture de sortie sont en effet calculés simultanément et indépendamment. Il est donc impossible, pour le *shader* responsable d'un *texel* donné, d'aller écrire sur un autre.

6.2.2 Bibliothèques logicielles

Les développements logiciels effectués lors de cette thèse se basent sur la bibliothèque *OpenGL*[8] et sur la suite logicielle *Cg*[66]. Mais la poursuite de ces travaux commanderait de travailler avec une autre suite logicielle, CUDA, destinée uniquement au calcul généraliste. Nous la présenterons à ce titre, à la suite des deux premières.

OpenGL est un ensemble de spécifications pour API [7] dédié au rendu graphique 2D et 3D. Son domaine d'application est potentiellement vaste : jeux vidéos, C.A.O., réalité virtuelle, etc. À l'origine en 1992, ces spécifications ont été développées par la société Silicon Graphics Industry et sont depuis contrôlées par Khronos Group, celui-ci succédant en 2006 à l'ARB (OpenGL *Architecture Review Board*).

Aujourd'hui, des bibliothèques *OpenGL* sont disponibles sur la plupart des plateformes matérielles (y compris récemment sur les téléphones mobiles) et les spécifications sont soutenues par bon nombre d'acteurs importants de l'industrie informatique : Apple, Sun, NVIDIA, etc. Microsoft, qui a décidé de quitter l'ARB en 1993, développe son propre système concurrent (Direct3D) pour le marché du jeux vidéo.

En pratique, *OpenGL* définit un ensemble d'environ 250 fonctions pour le dessin et le rendu de scènes 2D et 3D : formes, lumières, positionnement de caméra, transformations géométriques, etc.

Il existe une implémentation libre (au sens de la GPL, *General Public License*) d'OpenGL, Mesa3D [5], mais nous avons choisi d'utiliser la version fournie par NVIDIA pour deux raisons :

– La première est la disponibilité de plus de fonctions OpenGL. NVIDIA communiquant peu sur les spécificités de son matériel, il est naturellement difficile aux contributeurs du logiciel libre d'implémenter de manière optimale des fonctionnalités très récentes de la carte (décrites par une version récente d'*OpenGL*).

[7] *Application Programming Interface*, Interface de programmation d'application, bibliothèque logicielle

– La deuxième raison a trait à la stabilité de la bibliothèque : NVIDIA est obligée de fournir une bibliothèque de grande qualité car celle ci, seul intermédiaire entre le programmeur et la carte graphique, est garante de l'efficacité de la plateforme matérielle.

Dans nos travaux, nous utilisons *OpenGL* pour le rendu dans un écran virtuel (*off-screen buffer*) des résultats (y compris intermédiaires) de notre algorithme de recherche de plus proches voisins (cf. section 6.3). La figure 6.3 traduit le fonctionnement du procédé :

– On définit d'abord la **projection** : dans une scène 3D, une texture peut être projetée par exemple sur une des faces d'un cube. Nous choisissons ici de la projeter sur un quadrilatère de l'espace allant de l'origine $(0, 0)$ au point de coordonnée (S, S) (le quadrilatère a les mêmes dimensions que la texture).
– On définit ensuite l'**angle de vue**, c'est-à-dire la position et l'ouverture de la caméra. Celle-ci sera face au plan et cadrera exactement le quadrilatère projeté.
– On déclenche enfin le **rendu** de la scène. Il s'agit du plaquage de la texture (qui correspond, dans notre cas, au calcul) et de la prise de vue (production d'une texture sortie contenant les données).

La projection, l'angle de vue et le rendu sont à chaque fois définis ici de manière à traiter un quadrilatère de dimension (S, S). Cela permet de conserver pour chaque point (d'une texture en mémoire, d'une texture plaquée dans l'espace ou d'une texture filmée) les mêmes coordonnées allant de $(0, 0)$ à (S, S). En effet, si par exemple, une texture se voyait plaquée sur un quadrilatère de l'espace de dimension moindre, alors les texels seraient combinés de manière à réduire les dimensions de la texture. Or nous devons veiller à l'intégrité des données stockées ; une telle réduction (par des algorithmes dédiés au graphisme) est donc à proscrire.

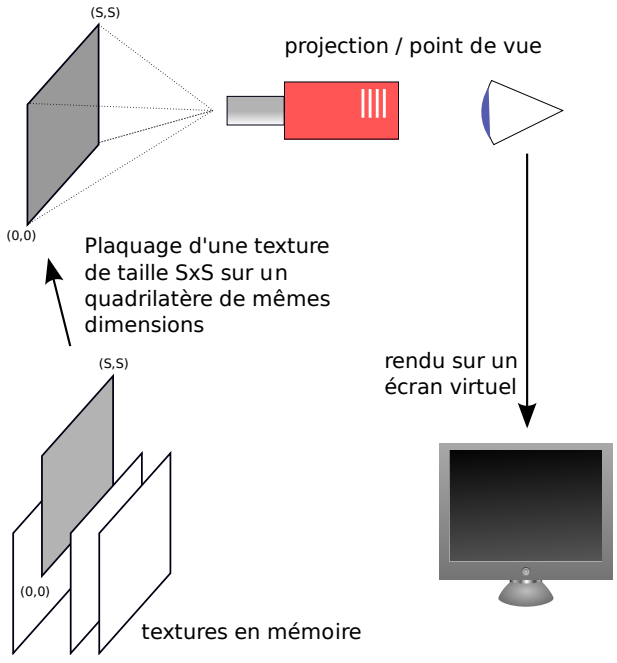


Fig. 6.3: Fonctionnement d'un calcul GPU en utilisant OpenGL

Cg. L'étape de rendu de la scène est en fait, dans notre cas, l'étape de calcul. Lors de la production de la texture finale, un programme *Cg* préalablement compilé est transféré au GPU puis exécuté.

Cg signifie *C for graphics* (cf. [66], [1]). C'est un langage dont le compilateur, fourni par NVIDIA, fabrique du langage machine adapté aux *shaders*. Ce langage est similaire au C mais comprend en plus des types de données adaptés aux GPU comme les *texels* et les textures.

CUDA. Comme il a été dit précédemment (cf. 6.1.2), la sortie en février 2007 de la bibliothèque CUDA (*Compute Unified Device Architecture* [2]), en même temps que l'architecture unifiée de la série 8 ainsi que les premiers calculateurs Tesla, marque un tournant dans le domaine du calcul généraliste sur GPU : dorénavant, la preuve de l'efficacité de ces systèmes est faite et les développements ne seront plus prospectifs ou expérimentaux, mais potentiellement professionnels (de nombreuses sociétés se sont créées en exploitant les GPU et CUDA pour, entre autres, le séquençage génétique, l'OPC (*Optical Proximity Correction*) en microélectronique, etc.)

La plateforme de développement CUDA est constituée d'un ensemble de bibliothèques C++ ainsi que d'un compilateur entièrement dédié à l'utilisation généraliste des GPU NVIDIA. Elle s'appuie plus fondamentalement sur l'architecture matérielle et s'éloigne de la vision restrictive texture-traitement d'image-projection sur l'écran. De plus, CUDA permet d'exploiter au mieux les nouveaux GPU de la marque et supprime certaines limitations : le calcul des nombres en double précision (64 bits) est désormais possible ainsi que l'écriture aléatoire en mémoire (*scattering*).

Les travaux de cette thèse concernant les GPU ont été pour une grande partie antérieurs à la sortie de CUDA et des nouvelles architectures unifiées ; les algorithmes ne sont donc pas optimisés pour les nouveaux GPU. Néanmoins, l'utilisation des bibliothèques *Cg* et *OpenGL* permet, comme on le verra dans la section 6.4, de satisfaire nos besoins en rapidité de calcul. Cependant, les développements futurs devront commencer par une migration complète des codes sources vers CUDA.

6.3 Recherche des k plus proches voisins par un GPU

L'avènement des GPU et leur capacité de calcul potentielle très élevée ont poussé certaines équipes à tenter de développer un moyen de résoudre le problème des k-P.P.V. (cf. chapitre 5) sur ce matériel. En 2006, Bustos *et al.*[15] proposent de trouver l'unique plus proche voisin en utilisant astucieusement la technique de *réduction* de textures. Les résultats, pourtant très prometteurs, n'auront pas de suite publiée. En 2008, les Français Garcia *et al.*[27] publient des résultats intéressants concernant la recherche de k-P.P.V. avec un GPU, mais, dans leur cas, l'architecture parallèle est utilisée pour traiter plusieurs problèmes simultanément (plusieurs requêtes) et non pas pour accélérer une seule requête qui serait acquise à chaque pas de temps.

Notre méthode est inspirée des travaux de B. Bustos pour les grands principes : calcul de distance, stockage d'indices et réduction. Elle tire néanmoins parti des nouvelles possibilités offertes par les cartes graphiques récentes, comme le rendu sur un écran virtuel. De plus, nous avons généralisé le problème à plusieurs P.P.V. en développant une nouvelle structuration des données.

La recherche des k-P.P.V. implique, en plus d'une structuration des données particulière, trois étapes de calcul et autant de programmes *Cg* :

1. Calcul des distances entre chaque élément de la bibliothèque et la signature acquise ;
2. Pré-calcul de minimums et stockage des indices ;
3. Réduction (conjointement à des calculs de minimums).

6.3.1 Représentation des données en mémoire

Comme pour la plupart des applications GPGPU conçues avec *Cg* et *OpenGL*, les données numériques sont représentées en mémoire sous forme de *textures*. Cette notion est héritée du graphisme et désigne un tableau de données considéré comme bi-dimensionnel. Les GPU sont capables de gérer différents types de textures et nous utiliserons ici un type destiné à stocker une image en couleur. Chaque élément de cette texture (*texel*) comportera donc quatre canaux : rouge, vert, bleu et α (transparence). Chaque canal de chaque *texel* peut stocker un mot de type `float` (nombre à virgule flottante de simple précision stocké sur 32 bits) en accord avec la norme IEEE 754.

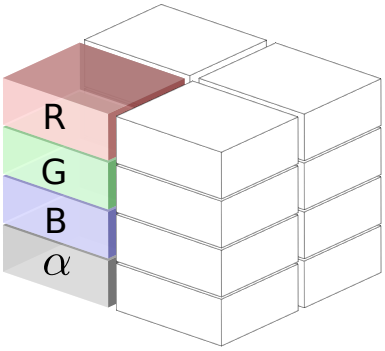


Fig. 6.4: Un élément de texture : *texel*

Puisqu'il s'agit de stocker la bibliothèque $\mathcal{B}$ dans la mémoire de la carte graphique, nous créons autant de textures que cette bibliothèque a de dimensions : $2N_\lambda$. Chaque signature $\mathbf{s}$ sera donc décomposée, et ses coordonnées (c'est-à-dire les deux signaux ellipsométriques choisis pour chacune des longueurs d'ondes utilisées) seront réparties dans des textures différentes, mais au même emplacement.

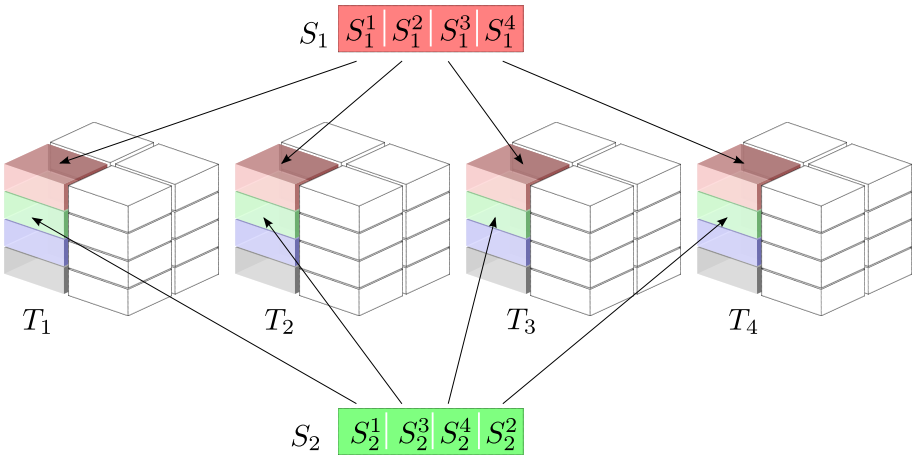


Fig. 6.5: Stockage de la bibliothèque en mémoire : éclatement en multiples textures

6.3.2 Calcul des distances

Une fois la base de donnée convenablement stockée dans la mémoire de la carte graphique, la recherche des k-P.P.V. commence par le calcul des distances entre la signature requête $\tilde{\mathbf{s}}$ et chaque signature de la bibliothèque $\mathcal{B}$. Cette opération étant commune à chacun des *texels*, elle est parallélisable et sera réalisée très efficacement par le GPU.

Un programme *Cg* par exemple (n'importe quel programme pour *shader*, que l'on appelle un *kernel*) dispose de **points d'attache** entre ses paramètres-textures d'entrées/sortie et les textures proprement dites stockées en mémoire (ceci peut être vu comme un mécanisme de pointeurs). Ce nombre de points d'attache, longtemps limité à 4, est aujourd'hui porté à 8 pour les GPU récents.

Ainsi, pour le calcul des distances, nous procéderons en plusieurs étapes. Sur la figure 6.6 est représenté le procédé pour un nombre de points d'attache égal à 4. Initialement :

- le point 0 (entrée) est attaché à la texture de calcul W_1, initialement nulle ;
- les points 1 et 2 (entrées) sont attachés aux textures des 2 premières coordonnées (que l'on appellera T_1 et T_2, et pour la suite T_i représentera la i$^{\text{e}}$ texture-coordonnées) ;
- le point 3 (sortie) est attaché à W_2, une autre texture de calcul.

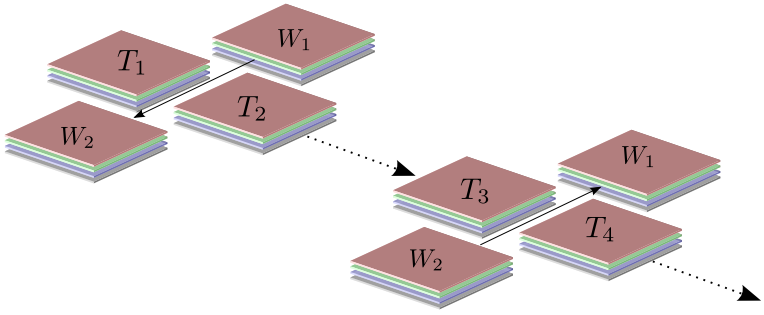


Fig. 6.6: Calcul des distances

Le calcul à effectuer sera :

$$W_2 \leftarrow W_1 + |\tilde{s}^1 - T_1| + |\tilde{s}^2 - T_2|$$

Ensuite, W_2 contenant des distances partielles sera attaché en entrée au point 0. Les deux textures des coordonnées suivantes seront attachées aux points 1 et 2 et W_1 servira de sortie attachée au point 3. Le calcul sera alors :

$$W_1 \leftarrow W_2 + |\tilde{s}^3 - T_3| + |\tilde{s}^4 - T_4|$$

Ceci sera répété jusqu'à obtenir dans l'une des textures W_1 ou W_2 les distances complètes. Chaque *texel* de la texture finale contient la distance entre la signature de la bibliothèque de même coordonnée et la signature acquise.

Ce procédé est appelé *ping-pong* dans la communauté du calcul GPGPU, les deux textures W_1 et W_2 se renvoyant l'une à l'autre les résultats intermédiaires.

6.3.3 Inclusion des indices

Lors du chargement en mémoire graphique de la bibliothèque, nous avons adjoint une texture de même taille et de même forme que les autres, contenant simplement les entiers 1 à $card(\mathcal{B})$.

Le rôle de cette texture est d'indexer chaque signature stockée, et donc chaque distance calculée ; cela permettra par la suite de garder une trace de chaque élément de la base et notamment de faire la correspondance entre le vecteur-signature stocké sur la carte graphique et le vecteur-paramètre stocké dans la mémoire vive de l'ordinateur. À la i^{e}position de la texture I (indices) est stocké l'entier i, correspondant à la i^{e}signature.

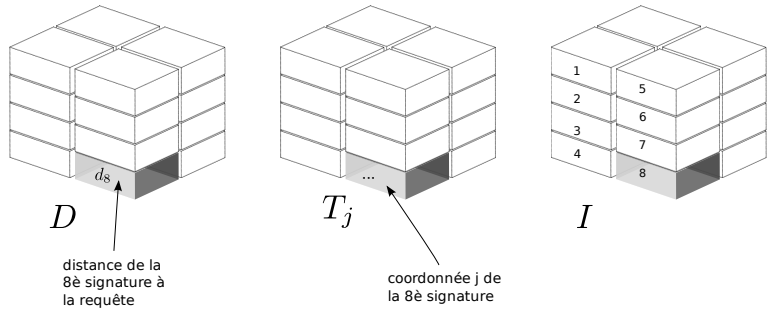


Fig. 6.7: Distances et indices sont stockés dans deux textures

La suite du procédé consistera à fusionner les textures D et I de manière à créer une nouvelle texture contenant à la fois distances et indices correspondants. C'est cette texture de taille initiale $4 \times S^2$ qui sera réduite pour fournir le résultat de la recherche.

La nouvelle texture sera structurée en *groupes* de texels. Chaque groupe devra avoir une taille suffisante pour stocker les k minimums et les k indices issus de chaque étape de la réduction. Ces groupes, de forme carrée, stockeront les minimums sur les deux couches supérieures (rouge et verte) et les indices sur les couches inférieures (bleue et α).

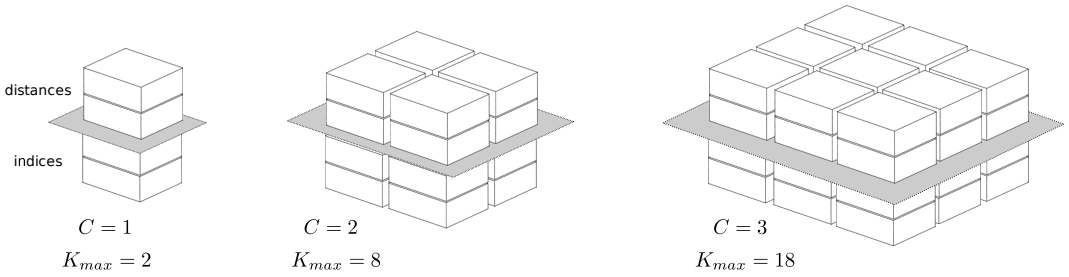


Fig. 6.8: Distances et indices sont agglomérés dans une texture structurée en groupes de texels de taille variable selon k

Ainsi, comme le montre 6.8, une largeur de groupe C permettra de trouver au maximum

$$k_{max} = 2 \times C^2$$

plus proches voisins.

6.3.4 Réduction

L'utilisation du principe de *réduction* est une conséquence directe d'une part du mode de fonctionnement parallèle du processeur graphique et d'autre part de l'interdiction d'écrire aléatoirement dans la mémoire. L'idée est que si l'issue d'un calcul doit être un seul élément, alors il est nécessaire de réduire en plusieurs étapes la quantité de données (potentiellement très grande) traitée en parallèle.

Un exemple simple d'utilisation de ce procédé serait une fonction *somme* qui calculerait simplement la somme des nombres stockés dans une texture. Dans chaque étape de la réduction, la texture est divisée en quatre parties de même forme, et chaque élément (*texel* ou groupe de *texels*) de la partie inférieure gauche par exemple recevrait la somme des 4 éléments situés à des endroits analogues sur chacune des parties (cf. figure 6.9). L'étape suivante serait identique sauf que la texture à considérer serait de taille quatre fois inférieure.

Pour notre application de recherche des k-P.P.V., cette technique de *réduction* permet d'obtenir au final le groupe de texels contenant les k-P.P.V. et leurs indices ; et cela en partant de la texture distances-indices. Si cette texture a une forme carrée de coté $S = C \times 2^n$, le procédé de réduction en entier se fera en seulement n étapes. À chaque étape, et pour chaque groupe de texels, le programme GPU (*kernel*) calcule les k-P.P.V. contenus dans les quatre groupes distants et les stocke dans le quart inférieur-gauche de la texture de sortie.

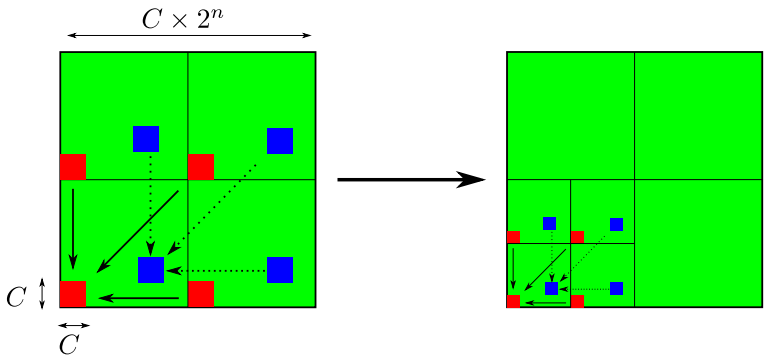


Fig. 6.9: Principe de *réduction* d'une texture. Le résultat est une division par 4 du nombre d'éléments de la texture

Taille de la bibliothèque. En considérant la structuration de la texture distances-indices en groupes de C texels d'une part et l'utilisation du procédé de réduction d'autre part, on déduit que la taille de la bibliothèque de signatures $\mathcal{B}$ doit être de la forme :

$$card(\mathcal{B}) = 4S^2 = 4C^2 \times 2^{2n}$$

Si la bibliothèque pré-calculée n'a pas cette taille, un bourrage sera opéré afin d'obtenir une taille adaptée. Ce bourrage consistera en des signatures impossibles, dont les chiffres sont très grands.

6.4 Performances des processeurs graphiques

Les résultats présentés dans cette section ont été obtenus avec la dernière carte graphique dont nous disposions. Cette carte comprend un GPU NVIDIA GTX 280 et une mémoire attachée de 1024Mio.

6.4.1 Éléments préalables à l'analyse des résultats

6.4.1.1 Indicateurs de performance

Nous avons choisi ici de mesurer la performance du processeur graphique de deux manières.

La première est une mesure absolue qui évalue la capacité de ce type d'architecture à résoudre rapidement un grand nombre de problèmes inverses par seconde (recherche des k-P.P.V.). Sur les graphiques concernés, on a tracé cette fréquence de résolution de problèmes inverses (notée IP/s), pour différentes valeurs de k, pour différentes tailles de bibliothèques ou encore pour différentes tailles de signatures.

La deuxième, mesure relative, estime la valeur ajoutée apportée par l'utilisation d'un processeur graphique. Il s'agit du gain en rapidité observé par rapport à un processeur scalaire d'ordinateur (CPU) pour l'exécution d'une recherche des k-P.P.V.. Ce gain, que nous appellerons par la suite **facteur d'accélération**, sera la valeur

$$a = \frac{t_{CPU}}{t_{GPU}}$$

obtenue pour un nombre donné (voir section suivante) de recherches consécutives de k-P.P.V..

Pour la comparaison nous opérerons cette recherche sur CPU Intel Xeon 3.0 GHz en utilisant l'algorithme de recherche de k-P.P.V. le plus efficace lorsque l'espace de recherche n'est pas structuré (cf. algorithme 1)[8]. Cet algorithme, qui parcourt de manière exhaustive les signatures de la bibliothèque, a un temps de résolution linéaire en $\mathcal{O}(nd)$, où n est la taille de la bibliothèque et d la dimension de chaque signature.

Algorithme 1

soit $\tilde{\mathbf{s}}$ la signature à comparer et soit $\mathcal{B} = \{\mathbf{s}_i\}_{1 \leq i \leq N}$ la bibliothèque de taille N.
pour i allant de 1 à k **faire**
  ajouter $\mathbf{s}_i$ au tableau des k-P.P.V.
fin pour
pour i allant de k+1 à N **faire**
  si $\|\tilde{\mathbf{s}} - \mathbf{s}_i\|$ est inférieur à l'une des k distances du tableau **alors**
    effacer le moins proche voisin du tableau
    mettre $\mathbf{s}_i$ à la place
  fin si
fin pour

6.4.1.2 Régimes transitoire et permanent du GPU

Les GPU sont construits pour les calculs graphiques intensifs de type *stream processing* ; ceci fait que ce type d'architecture atteint son rendement maximal quand les données sont disponibles et quand toutes les unités de calcul sont actives. En pratique, pour notre problème, il s'avère que les meilleures performances sont atteintes pour des recherches multiples dans une même base de données. Si l'on trace par exemple la performance d'une recherche de 8-P.P.V. dans les conditions habituelles de la scatterométrie, c'est-à-dire pour un vecteur de taille 32 (donc, 16 longueurs d'ondes) et une bibliothèque d'environ 4,2 millions de signatures ($S = 1024$, 512 Mio en mémoire), on obtient alors une évolution tracée sur la figure 6.10.

On remarque alors que la performance maximale est atteinte pour approximativement 100 recherches consécutives dans une même bibliothèque. Ce nombre sera utilisé pour les résultats qui vont suivre.

[8] La plupart du temps, pour effectuer des recherches de plus proches voisins, on procède à une structuration préalable de l'espace (en arbre, par exemple). Même si cela coûte un certain temps au départ, la recherche qui s'ensuit se trouve grandement accélérée.

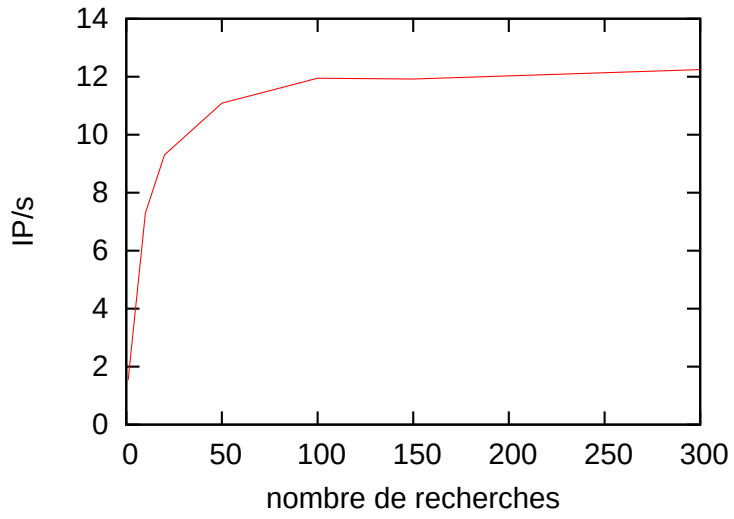


Fig. 6.10: Efficacité du GPU (en nombre de problèmes inverses par seconde) selon le nombre de répétitions. Courbe tracée pour 8-P.P.V.

Ceci ne constitue pas un problème pour notre application temps réel : ce "régime transitoire" est dû à un laps de temps incompressible qui apparaît lors de l'établissement de la première recherche [9]. Donc, à condition de mettre en route ce moteur de recherche GPU préalablement au procédé à suivre, on peut faire disparaître l'effet de ce temps. En pratique, si le système de reconstruction de profil est relié à un ellipsomètre qui débite 10 signatures par seconde, alors il conviendra de démarrer le programme une dizaine de secondes en avance.

6.4.1.3 Temps de transfert des données

La bibliothèque de signatures scatterométriques est calculée au préalable et se trouve ainsi stockée à un moment donné dans la mémoire de l'ordinateur. Pour réaliser une recherche de signature en utilisant le GPU, un transfert de données doit donc être effectué vers la mémoire de la carte graphique.

Vu les tailles de bibliothèques habituellement utilisées, ce temps de transfert n'est pas négligeable. Cela ne constitue pas un problème si une même bibliothèque est utilisée durant tout le suivi de procédé, mais en revanche, il peut arriver qu'on ait besoin de changer de bibliothèque pour, par exemple, en utiliser une plus raffinée ou mettre en place un autre modèle géométrique. Dans ce cas, on ne peut plus négliger ce temps de chargement (ni d'ailleurs le temps transitoire de la section précédente).

Nous avons tracé sur la figure 6.11 ce temps de transfert en fonction de la taille de la bibliothèque (en Mio) : nous avons gardé une taille de bibliothèque constante (4.2 millions de signatures, $S = 1024$) mais avons fait varier la largeur de chaque signature (deux fois le nombre de longueurs d'ondes). Cette manière de faire n'influe pas sur le débit mesuré.

Pour des tailles inférieures à 800Mio, on remarque que le comportement est linéaire et traduit un débit d'environ 600 Mio/s. Ce débit est loin de la vitesse théorique du bus liant reliant la carte à l'ordinateur (bus de type PCI-Express 16x : 4Gio/s) mais il est significatif aussi des opérations d'allocation et d'organisation de mémoire de la carte graphique.

[9] On parlerait d'*overhead*, en jargon angliciste.

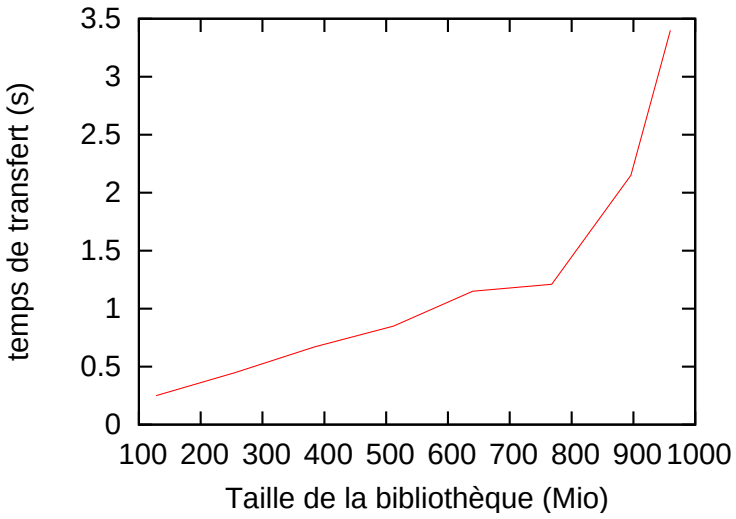


Fig. 6.11: Temps de transfert entre la mémoire centrale et la carte graphique

Les mesures reportées sur le graphique 6.11 sont difficiles à obtenir avec précision (nous avons représenté une moyenne) car elles dépendent des ressources processeur et mémoire de l'ordinateur hôte. Les conditions "expérimentales" sont difficiles à reproduire étant donné, par exemple, les nombreuses activités du système d'exploitation.

6.4.1.4 Limitations

Lors des tests et des mesures de performance de la plateforme GPU dont nous disposions (NVIDIA GTX 280 montée sur une carte GigaByte), il apparaît deux limitations notables :

- La quantité de mémoire est de 1 Gio et ne peut être étendue. De plus, dans notre cas, elle n'est pas utilisable en entier car la carte est utilisée en parallèle pour la gestion des fenêtres de l'interface graphique et l'affichage sur écran. Ceci a pour conséquence que les résultats et notamment les courbes présentés par la suite apparaîtront tronqués.
- Nous n'avons pas pu obtenir une recherche de plus de 13-PPV, alors que le programme transféré à la carte graphique est formellement identique pour des valeurs de k allant de 9 à 18 ($C = 3$). Ce problème peut avoir pour origine la conception du circuit GPU (en dépassant $k = 13$, on peut dépasser la capacité d'un registre, d'une fonction logique) ou à un *bug* dans la chaîne compilateur-assembleur de *Cg*. Des investigations pourraient être entreprises pour résoudre cela, mais il serait plus judicieux de migrer les codes sources en CUDA.

6.4.2 Mesures de performances

6.4.2.1 Application à la scatterométrie

Les signatures de scatterométrie acquises en temps réel avec un ellipsomètre multi-voies comportent au plus quelques dizaines de longueurs d'ondes. Les ellipsomètres dont nous disposions (Horiba/Jobin-Yvon UVISEL, cf 2.2) disposent par exemple de 16 voies pour le modèle situé sur le bâti de gravure et de 32 voies pour le modèle utilisé pour le fluage de résine. À chaque longueur d'onde est affecté deux signaux, en général il s'agit de I_S et I_C, (voir le chapitre 2 consacré à l'ellipsométrie) et, en mode dynamique, l'ellipsomètre débite ces couples de grandeurs à un rythme

maximal de 10 à 15 par seconde.

En conséquence, les bibliothèques à charger dans la mémoire de la carte graphique seront de dimensions 32 et 64, et l'intérêt de l'utilisation d'un GPU devra être démontré par sa capacité à traiter en temps réel le rythme des données fournies par l'ellipsomètre. Les résultats compilés dans les tableaux 6.3 représentent le rythme de traitement des signatures acquises, en nombre de problèmes inverses résolus par seconde (IP/s), pour diverses grandeurs de bibliothèques[10] et pour des dimensions de signature de 32 et 64 éléments. Par exemple, la recherche des 8 plus proches voisins d'une signature de 16 longueurs d'ondes (donc 32 dimensions) dans une bibliothèque d'un million d'éléments (S=512) se fera au rythme de 8.1 par secondes sur un CPU, et 48,5 par seconde sur un GPU. Dans ce cas, le processeur de la carte graphique apporte un gain de performance de 6.

2-PPV

32 dim./16λ	256	512	1024
CPU	32.8	8.1	2
GPU	185.2	48.3	11.8
a	5.6	5.9	5.8

64 dim./32λ	256	512
CPU	16.3	4.1
GPU	138.9	40.1
a	8.5	10

8-PPV

32 dim./16λ	256	512	1024
CPU	32.8	8.1	2
GPU	178.6	48.5	12.2
a	5.4	6	6

64 dim./32λ	256	512
CPU	15.2	3.8
GPU	119	40
a	7.8	10.6

12-PPV

32 dim./16λ	192	384	768
CPU	43.3	10.9	2.7
GPU	101	48.3	15.2
a	2.3	4.4	5.6

64 dim./32λ	192	384	768
CPU	27	6.8	1.7
GPU	87	43.3	13.7
a	3.2	6.4	8.1

Tab. 6.3: Nombre de problèmes inverses résolus par seconde puis facteur d'accélération *a* pour une application scatterométrie (16 et 32 longueurs d'ondes). Les valeurs grisées correspondent à des rythmes de traitement inférieurs à 10, c'est à dire incompatibles avec le débit de l'ellipsomètre.

À la lecture de ces tableaux, l'utilisation de la solution GPU se justifie dès que la taille de la bibliothèque dépasse le demi-million d'éléments, c'est à dire pour S=512 ou S=384. En effet, si l'on considère qu'un ellipsomètre fournit de l'ordre d'une dizaine de signatures par seconde, alors un système de traitement résolvant moins de 10 problèmes inverses par seconde serait inadapté. L'utilisation du GPU apporte dans tous les cas présentés ici un gain de vitesse compris entre 5 et 10 et rend possible le traitement de signatures débitées à la fréquence de l'ellipsomètre.

[10] On rappelle que la taille d'une bibliothèque, en nombre de signatures, est de la forme $card(\mathcal{B}) = 4 \times S^2 = 4 \times C \times 2^{2n}$ où C est la taille d'une cellule élémentaire (cf. section 6.3) ; les trois cas présentés correspondent donc à des valeurs de C de 1 (2-PPV), 2 (8-PPV) puis 3 (12-PPV). On a détaillé dans le tableau suivant les tailles de bibliothèques utilisées en fonction de C et S.

C	S	$card(\mathcal{B})$
1, 2	64	16 384
	128	65 536
	256	262 144
	512	1 048 576
	1024	4 194 304

C	S	$card(\mathcal{B})$
3	96	36 864
	192	147 456
	384	589 824
	768	2 359 296

6.4.2.2 Performances brutes

La technique de recherche de plus proches voisins en utilisant un processeur graphique admet d'autres domaines d'applications possibles que la scatterométrie, et ces domaines ne sont pas nécessairement caractérisés par une taille de signature aussi faible que 32 ou 64. Nous allons ici présenter les gains en rapidité pour des tailles de signature quelconques.

Les graphiques 6.12 représentent les facteurs d'accélérations pour des tailles de signatures exponentiellement croissantes. Chaque courbe correspond à une taille de bibliothèque différente en nombre de signatures.

Sur ces graphiques, on remarque que le facteur d'accélération a est d'autant plus grand :
- que la dimension de la signature est élevée.
- que la taille de la bibliothèque (en nombre de signatures) est grande.

Ces graphiques rendent compte de la plus value offerte par l'utilisation d'un processeur graphique mais n'indiquent pas la fréquence de résolution des problèmes inverses. Le tableau 6.4 présente un certain nombre de valeurs (en nombre de problèmes inverses résolus par seconde) pour différentes conditions (taille de signatures, nombre k) et pour une taille de bibliothèque maximale, ceci afin de donner une idée du potentiel des GPU.

S	**dimension**	**taille** (Mio)	**2-PPV**	*a*	**8-PPV**	*a*
128	2000	500	12.4	*5.8*	12.5	*5.9*
256	800	800	24.5	*18.3*	24.2	*18.4*
512	200	800	19.8	*15*	22.1	*17*
1024	50	800	9.9	*7.6*	10.3	*9.1*

			12-PPV	*a*
192	1500	843.75	12.7	*10.25*
384	400	900	15.2	*13*
768	100	900	10.2	*9*

Tab. 6.4: Nombre de problèmes inverses résolus par seconde pour différentes valeurs de k et pour des tailles de bibliothèques maximales.

6.5 Conclusion

L'efficacité de l'algorithme de reconstruction présenté dans le chapitre 5 pour une application temps réel est conditionné par la rapidité à laquelle la recherche des plus proches voisins dans la bibliothèque est effectuée. Pour des bibliothèques de taille supérieure au demi-Gio, la technique simple de recherche par CPU n'est plus valable ; elle se heurte en effet au rythme d'acquisition des signatures qui est, rappelons-le, de l'ordre d'une dizaine par seconde.

L'utilisation d'un processeur graphique résout de manière franche ce problème : la puissance qu'offre son architecture vectorielle pour le problème des k plus proches voisins permet de changer le facteur limitant du système de suivi de procédé pris dans son ensemble ; c'était la puissance de calcul, c'est désormais la rapidité de l'ellipsomètre à acquérir des signatures. Ceci semble avoir un caractère définif tant l'accroissement des performances des GPU au cours du temps est considérable et tant il est difficile de concevoir un ellipsomètre beaucoup plus rapide.

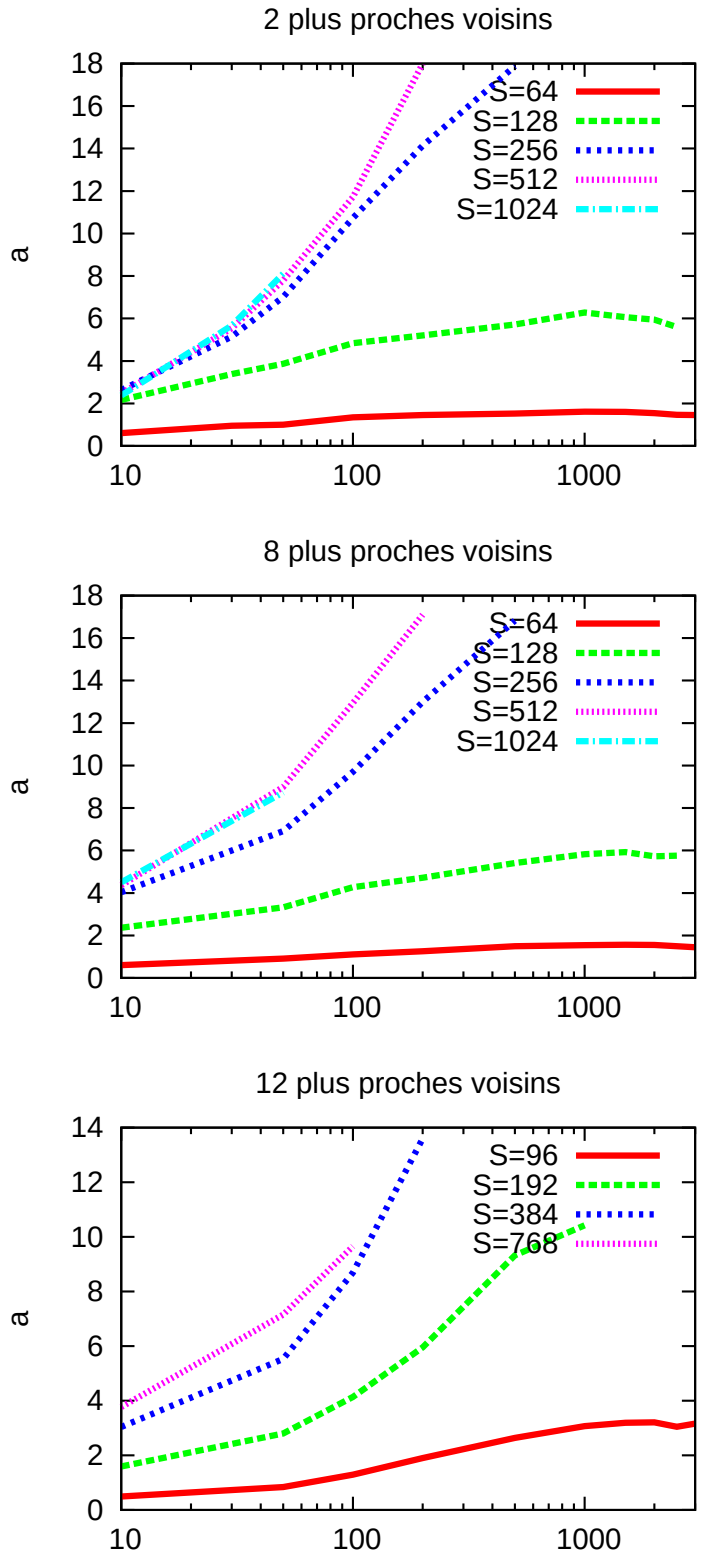


Fig. 6.12: Évolution du facteur d'accélération a en fonction de la taille de la signature pour différentes grandeurs de bibliothèques (S) et pour différentes valeurs de k

Troisième partie

Suivi en temps réel de procédés

Dans cette partie seront exposés trois exemples de suivi de procédés dynamiques. Pendant ces travaux de thèse, ces cas ont servi à élaborer dans un premier temps, puis à valider dans un second, notre méthode de reconstruction de paramètres géométriques exposée au chapitre 5. Le premier de ces cas est purement artificiel et vise à montrer le mode d'action du procédé. Les deuxième et troisième cas sont, au contraire, expérimentaux : il s'agit du suivi de la forme que prend un créneau de résine lorsqu'il se déforme sous l'action de la chaleur (fluage) ou d'un procédé plasma (gravure).

CHAPITRE 7

Exemple simulé

Un exemple démonstratif de la méthode de reconstruction est donné ici par une expérience purement artificielle, fondée sur la simulation de l'évolution temporelle de caractéristiques physiques.

7.1 Génération du film de signatures

Soit un système diffractant consitué d'un réseau de lignes de résine reposant sur un substrat de silicium. On considère que la forme du motif est trapézoïdale donc modélisée par trois paramètres géométriques (voir la figure 7.1) :

- CD : il s'agit de la largeur à mi-hauteur du motif (*Critical Dimension*, dimension critique).
- H : hauteur de résine.
- α : angle de l'inclinaison des flans (souvent dénommé *SWA, sidewall angle*)

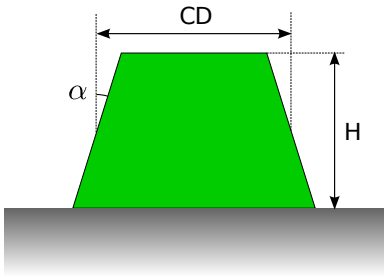
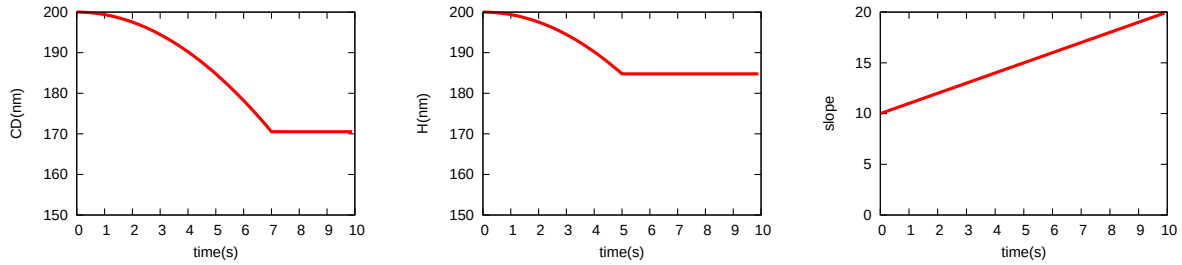


Fig. 7.1: Modélisation du créneau par trois paramètres

Par un procédé quelconque (on peut penser à un procédé de gravure plasma, cf. chapitre 9), on considère que les trois grandeurs précédentes varient en fonction du temps de la manière tracée sur la figure 7.2.

Cette expérience est certes construite par simulation, mais elle correspond typiquement à ce que l'on souhaite mesurer en microélectronique ; beaucoup de procédés sont mis au point très finement de manière à contrôler au maximum ces trois grandeurs.

La variation des paramètres CD et H comporte, on le voit, une partie régulière, une cassure, et une partie finale plate.



(a) CD (b) H (c) α

Fig. 7.2: Variation des paramètres géométriques du profil de résine

En utilisant un code de calcul électromagnétique (il s'agit du problème direct, cf. 3.3), on calcule l'ensemble des signatures scatterométriques engendrées par un tel profil : une signature est générée à chaque pas de temps (100ms). Ces signatures sont ici relatives au couple (I_S, I_C), mais un choix différent n'altérerait pas la démonstration.

Sur chacune de ces signatures, nous ajoutons un bruit de nature uniforme et d'amplitude I :

$$S' = S + I(1 - 2\theta), \theta \in [0; 1]$$

de telle sorte que S' varie aléatoirement entre $S+I$ et $S-I$. Ce facteur aléatoire est supposé englober pour ce cas tous les types d'erreurs de nature expérimentale. Nous prendrons une valeur $I = 0.1$; cela est largement surévalué mais contribue à l'illustration du procédé de reconstruction. Sur la figure 7.3 est représenté le changement dans la forme d'une signature dû à l'ajout de bruit.

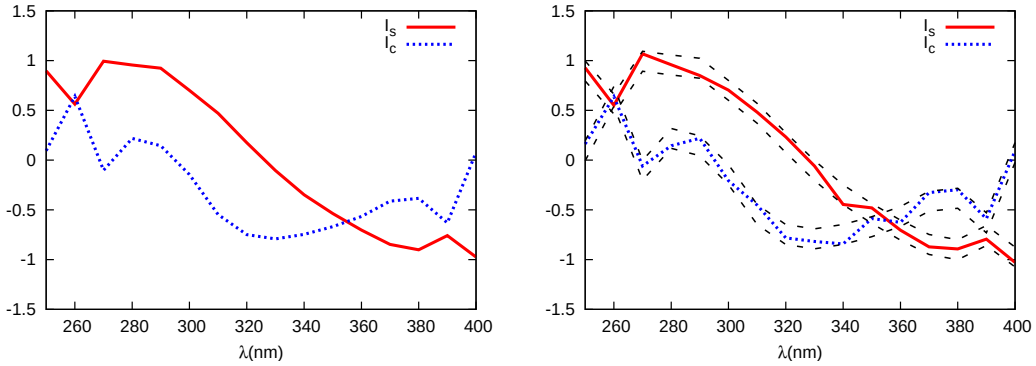


Fig. 7.3: Ajout d'une composante aléatoire à chaque signature. Les courbes en pointillés qui encadrent la signature de droite représentent l'amplitude I avec laquelle chaque point de la signature d'origine peut varier.

7.2 Reconstruction du profil

À ce point, il reste à résoudre le **problème inverse dynamique** : nous créons pour cela une bibliothèque de 4096 éléments (37 minutes sur une machine quadricoeur). Les valeurs prises par les paramètres sont données dans le tableau suivant :

	min	pas	max
CD (nm)	161.6	2.56	201
H (nm)	161.6	2.56	201
α (degrés)	9.5	0.7	20.1
période	400		

Comme il est détaillé dans le chapitre 5 traitant de la reconstruction dynamique de profil, tout commence par la recherche dans la bibliothèque des plus proches voisins de chaque signature. Pour cet exemple, nous prendrons $k = 4$: quatre jeux de paramètres (CD, H, α) seront donc extraits, à chaque pas de temps, de la recherche dans la bibliothèque. Les croix rouges des graphes de gauche de la figure 7.5 représentent les **projections** sur les plans (t, CD), (t, H) et (t, α) de ces quatre jeux.

L'étape suivante est le choix d'une seule signature parmi les 4-PPV. Le jeu de paramètres choisi est représenté (en projection toujours) sur les mêmes graphiques par une ligne continue rouge. Il est remarquable ici que, pris isolément, la variation d'un seul paramètre peut s'avérer irrégulière (plus irrégulière que ce qu'implique la nature discrète de l'espace). Cette étape impose en effet la régularité du **vecteur paramètre**, pris dans sa totalité.

Enfin, après l'étape finale de lissage (régularisation de Tikhonov), on obtient les courbes rouges des graphiques de droite de la figure 7.5. Celles-ci doivent être comparées aux courbes en pointillés bleus (qui sont les courbes ciblées, cf. fig. 7.2). On a aussi tracé, en $+$ rouges (**raw**, brut), ce qu'aurait été la variation du profil sans ce procédé de reconstruction, avec une simple recherche dans la bibliothèque : particulièrement irrégulière.

On a ainsi montré l'efficacité du procédé de reconstruction qui a été développé durant cette thèse.

7.3 Remarque : Justification empirique de la norme 1

Au chapitre 4, nous avons affirmé que, ne connaissant rien de la statistique associée aux résidus (cf. éq. 4.1), l'utilisation de la norme 1 se justifiait d'avantage que la classique méthode du χ^2. En voici une illustration : nous avons comparé la reconstruction précédemment effectuée avec la norme 1 avec ce qu'elle aurait été en utilisant la norme 2.

Pour ne pas alourdir ce document avec des figures menant à des conclusions redondantes, nous avons choisi de ne représenter sur les figures 7.5 que les variations du CD (largeur du créneau).

On observe que :

- Sans bruit ajouté au film de signatures scatterométriques, la reconstruction faite avec la norme 2 introduit un biais. Même si l'on a forcé le CD initial à 200nm, on remarque qu'il redescend tout de suite à environ 3 nm en dessous de la courbe attendue. **On montre là l'erreur introduite par la nature discrète de la bibliothèque de signatures** ; c'est en effet ici la seule source d'erreur puisque le calcul des signatures du film est effectué dans les même conditions que le calcul de la bibliothèque. La comparaison avec la norme 1 montre clairement la plus grande robustesse de celle-ci vis-à-vis de cette erreur due à la bibliothèque.
- L'ajout du bruit ne change rien à cette observation ; le biais est toujours présent.
- Si le coin dans les variations à $t = 7s$ est adouci par la norme 1, il semble disparaître complètement avec la norme 2.

Nous justifions ainsi empiriquement ce qui rend la norme 1 préférable à la norme 2, **dans les conditions choisies pour résoudre notre problème inverse**, à savoir : calcul électromagnétique pour le problème direct, puis méthode des bibliothèques pour le problème inverse.

En conclusion, nous avons cherché à montrer ici le mode d'action de l'algorithme de reconstruction de paramètres géométriques que nous avons développé. Son efficacité parait satisfaisante pour

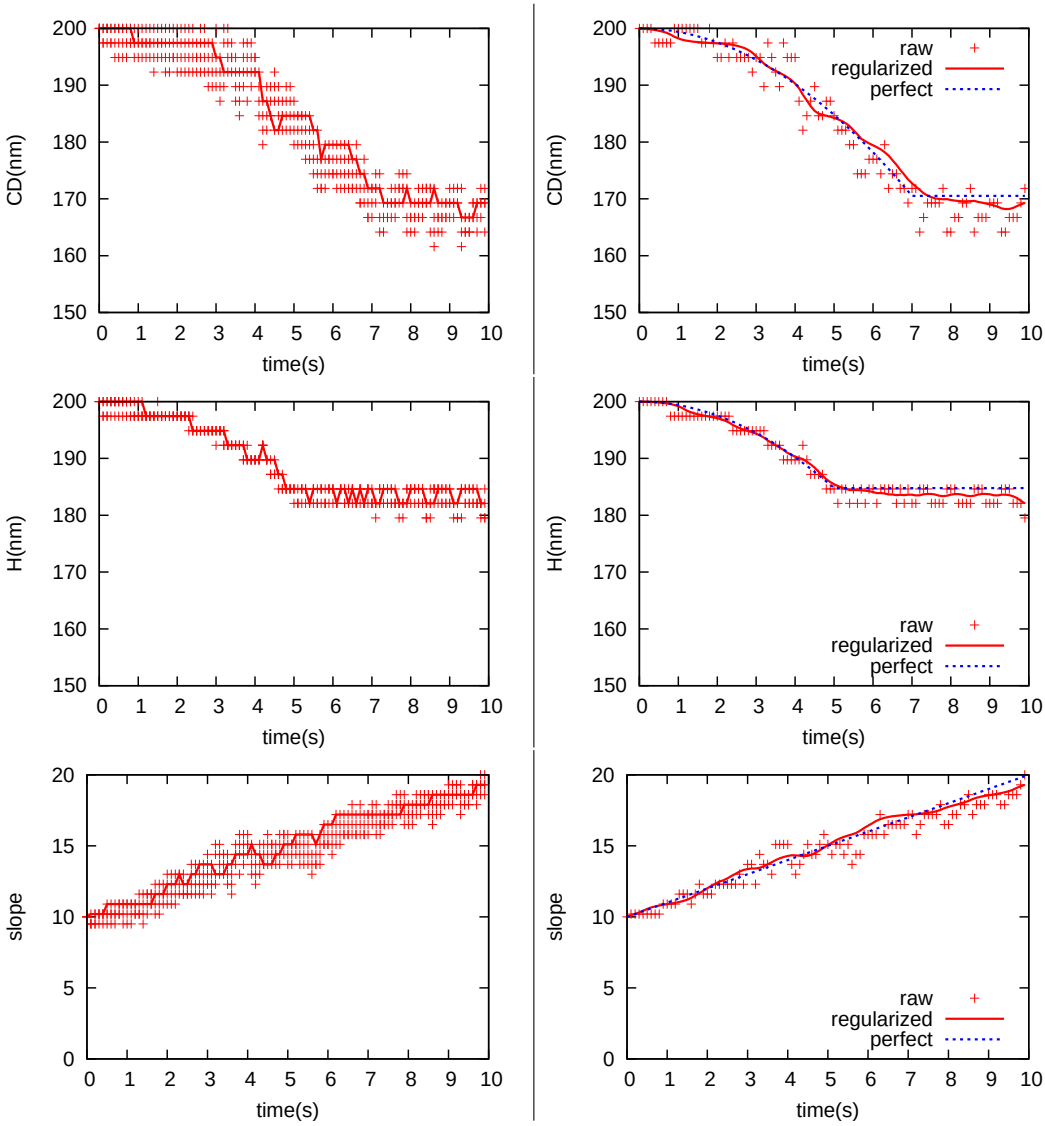
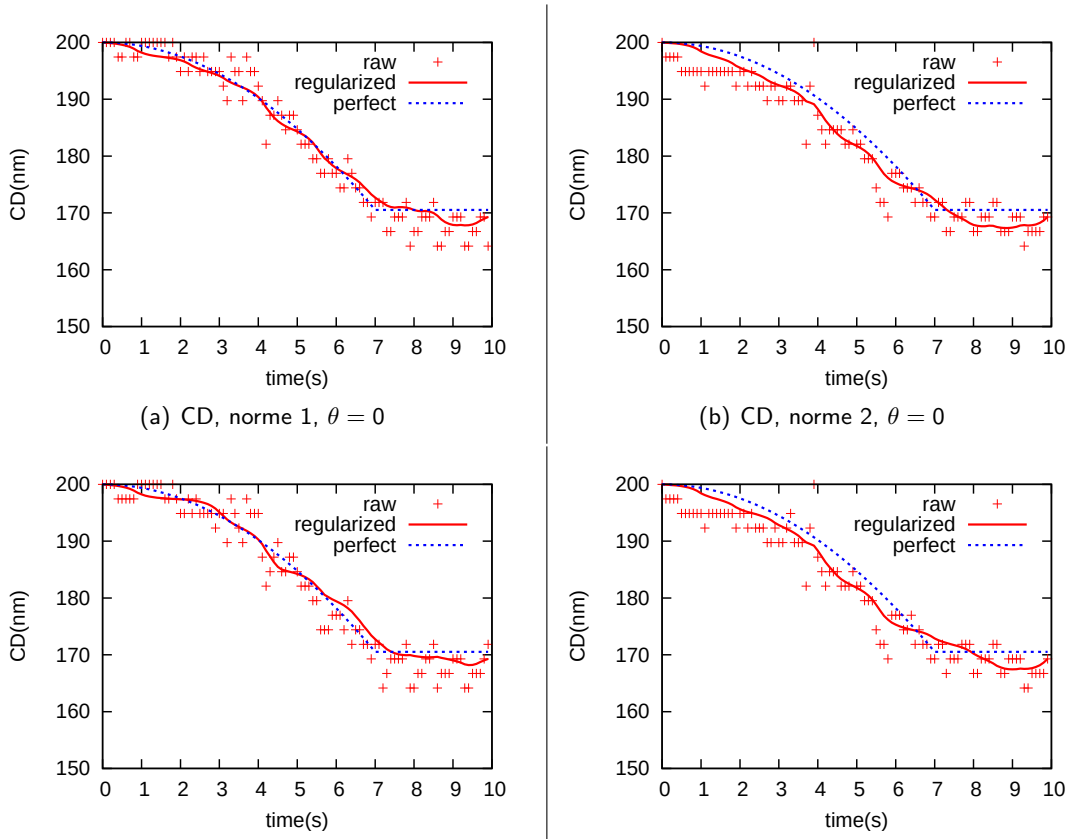


Fig. 7.4: Reconstruction dynamique des paramètres géométriques CD, H, α : dans les trois graphiques de la colonne de gauche, les croix représentent les k-P.P.V.issus de la bibliothèque, et la courbe rouge le résultat de la sélection. Dans la colonne de droite se trouve le résultat final de la reconstruction, comparé aux variations attendues.



(a) CD, norme 1, $\theta = 0$

(b) CD, norme 2, $\theta = 0$

(c) CD, norme 1, $\theta = 0.1$

(d) CD, norme 2, $\theta = 0.1$

Fig. 7.5: Reconstructions comparées du CD en utilisant les normes 1 et 2, avec et sans bruit

compenser l'erreur introduite par la méthode des bibliothèques pourvu que le modèle géométrique soit adapté au procédé et que les paramètres de régularisation soient bien choisis. En effet, pour ces derniers, c'est la connaissance *a priori* du procédé qui permettra d'obtenir le suivi voulu : si par exemple, le procédé est susceptible de varier brusquement (cf. coin à $t = 7s$), alors une trop forte régularisation est à proscrire. Mais au contraire, si on le sait linéaire ou variant très régulièrement, alors on pourra lisser plus franchement.

CHAPITRE 8

Fluage de résine

Le développement de notre méthode de reconstruction ainsi que sa première validation expérimentale se sont faits conjointement aux travaux de T. Leveder [54] sur l'optimisation du procédé de nano-impression thermique.

Ce procédé de lithographie (Chou *et al.* [21]) consiste à fabriquer des motifs en appliquant un moule dur (plaque de silicium, ce matériau supporte la montée en température et sa gravure est bien maîtrisée) sur une couche plane de résine. Cette méthode, souvent désignée par l'acronyme NIL (*Nano-Imprint Lithography*) est de nos jours validée en contexte industriel pour la fabrication de motifs de haute résolution (nanométriques), mais souffre encore de problèmes de défauts et de débit de production [41].

En pratique, l'étape de lithographie par nano-impression consiste à chauffer la résine au delà de sa température de transition vitreuse[1] pour la rendre visqueuse, la presser pour qu'elle prenne la forme du moule, la refroidir pour figer les motifs et, enfin, la démouler (cf. fig. 8.1).

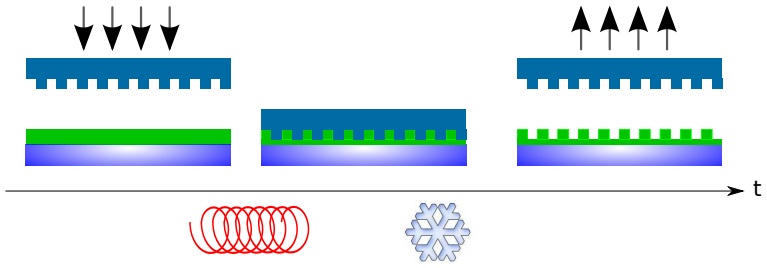


Fig. 8.1: Procédé de lithographie par nano-impression : un moule dur de silicium est plaqué sur une couche plane de résine pour former des motifs. La presse utilisée est une EVG 520ED ; elle autorise une pression uniforme de 40kN sur une plaque de 200mm et un vide de 5×10^{-4} bar.

Les étapes de chauffage et refroidissement prennent un temps considérable lors de la fabrication des motifs. Les travaux de T. Leveder ont pour ambition d'optimiser les températures de fluage afin d'améliorer le remplissage du moule mais aussi d'accroître le rendement de production. La

[1] Il s'agit de la température en dessous de laquelle un matériau ne se déforme pas.

compréhension du comportement des résines aux abords de leur T_g (température de transition vitreuse) a de plus un intérêt fondamental pour la science des matériaux[2].

Un motif nano-imprimé est mis à fluer par une température supérieure à T_g. Il évolue dans le temps avec une vitesse issue d'un équilibre entre la viscosité dynamique η qui tend à freiner le fluage et l'énergie de surface σ qui provoque ce fluage (l'évolution tend en effet à minimiser cette énergie). Cela se retrouve dans le dimensionnement du rapport de ces grandeurs :

$$\frac{[\sigma]}{[\eta]} = \frac{J.m^{-2}}{Pa.s} = \frac{N.m^{-1}}{N.s.m^{-2}} = m.s^{-1}$$

Or, si l'énergie de surface est connue pour varier peu dans la gamme de température que l'on utilise, il en est autrement de la viscosité dynamique. Celle-ci dépend de la température T par la relation suivante ([54] ch. 5) :

$$\eta(T) = \frac{(2\pi)^4}{3} \cdot \sigma \cdot \tau_1(T) \cdot \frac{h_0^3}{\lambda^4} \tag{8.1}$$

où λ est la largeur du motif (période du réseau dans notre cas), h_0 sa hauteur moyenne et $\tau_1(T)$ un temps de relaxation caractéristique de la décroissance du motif. C'est ce temps que nous allons chercher à mesurer.

Pour cela, nous avons suivi dynamiquement le profil d'un réseau de lignes fabriqué par nano-impression lors d'une mise à température T ($T > T_g$ pour permettre le fluage) sur une plaque chauffante (figure 8.2).

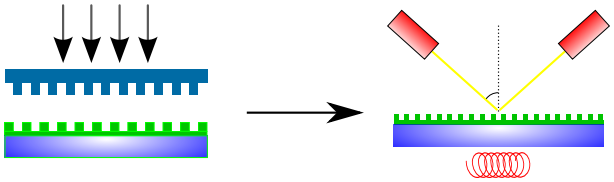


Fig. 8.2: Dispositif expérimental dédié au fluage de résine.

T. Leveder a montré à l'aide des équations de la mécanique des fluides [51] que la variation de forme d'un créneau de résine qui se met à fluer pouvait se décomposer analytiquement en série de Fourier :

$$h(x,t) = h_0 + \sum_{n=1}^{\infty} H_n e^{-\frac{t}{\tau_n}} \cos\left(\frac{2\pi n x}{\Lambda_x}\right) \tag{8.2}$$

Chaque ordre flue à une vitesse différente et avec une dépendance en temps exponentielle dont le temps caractéristique est, pour le premier ordre, la grandeur $\tau_1(T)$ que l'on cherche.

Les mesures faites par microscopie à force atomique sur des créneaux de résine polystyrène-130K (masse molaire : 130kg/mol) mis à fluer au delà de la température de transition vitreuse ($T_g = 100° \pm 1$) semblent accréditer le modèle (cf. fig. 8.3) : apparitions d'instabilités de Gibbs (petites bosses sur la deuxième image) et comportement sinusoïdal aux temps longs (les ordres élevés semblent disparaître en premier).

[2] On notera qu'il n'existe aucun moyen pour mesurer la T_g d'un film mince sur son substrat, ce qui est pourtant un besoin fondamental.

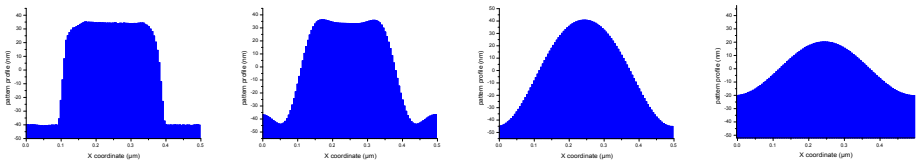


Fig. 8.3: Déformation d'un créneau de résine PS130K durant son élévation de température. Mesures réalisées par AFM.

Si l'on utilise ce modèle pour la génération d'une bibliothèque, la scatterométrie dynamique doit permettre d'en déterminer certaines constantes, en particulier la constante τ_1 qui est liée à la viscosité (éq. 8.1). Il est en effet possible de comparer l'amplitude du premier ordre de la décomposition avec une mesure AFM : il s'agit (au premier ordre) de la hauteur maximale du créneau de résine :

$$A_1(t) = H_1 e^{-\frac{t}{\tau_1}} \tag{8.3}$$

Pour cela nous avons créé une bibliothèque particulière dont la construction suppose d'utiliser la forme initiale mesurée par AFM, $h(x, t_0)$, et de prévoir son évolution dans le temps. Cette bibliothèque comporte toujours un grand nombre de géométries, mais, cette fois, l'indexation est faite par un "temps fictif" t_f qui correspond à un τ unitaire. Plus précisément, comme ces géométries sont supposées varier exponentiellement avec le temps ; la bibliothèque est indexée par e^{t_f}.

Nous avons représenté sur le graphique 8.4 le suivi dynamique du fluage de notre réseau de polystyrène (période : 500nm, hauteur moyenne : 180nm) porté à 130°. Nous avons suivi, en pratique, la grandeur e^{t_f} en fonction du temps, et les mesures issues de la scatterométrie sont comparées aux mesures AFM.

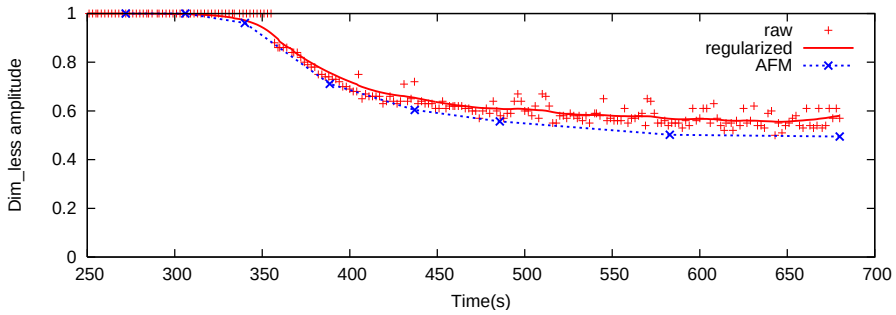


Fig. 8.4: Variation de forme d'un créneau de résine en fonction du temps. On a représenté ici la valeur relative du premier ordre de la déformation (e^{t_f}) en fonction du temps. Les résultats issus de la scatterométrie sont en rouge ; brut pour les croix, et régularisé pour la courbe. Les croix bleues correspondent aux relevés AFM.

Bien qu'elles aient le même comportement, on peut remarquer à la lecture de ces courbes que les mesures AFM semblent s'éloigner au cours du temps des mesures scatterométriques. Plusieurs éléments pourraient causer cela :

– Les courbes de dispersion (indices optiques) du matériau sont susceptibles de bouger au cours du temps à cause des phénomènes de dilatation ou au contraire de réarrangement des macro-

molécules au sein du polymère. Or cela n'est pas pris en compte lors de la création de la bibliothèque.
- L'utilisation de la MMFE implique une géométrie découpée en couches. Or une modélisation correcte de certains détails (comme les petites bosses qui apparaissent en début de fluage, cf. 2e figure de 8.3) impliquerait un trop grand nombre de couches, donc une convergence difficile de l'algorithme. Du fait même de cette décomposition la MMFE n'apparaît donc pas comme la méthode la mieux adaptée pour la modélisation des formes sinusoïdales de faible amplitude. L'utilisation d'un autre algorithme, comme la méthode C ([20], [61]), serait ici plus judicieux.

Néanmoins la variation exponentielle est visible, et on peut ainsi retirer de ces courbes la grandeur τ_1, temps caractéristique du fluage et grandeur reliée à la viscosité dynamique de la résine (éq. 8.1) :

$$e^{t_f} = e^{-t/\tau_1} \Rightarrow \tau_1 \sim 100s$$

Pour une application numérique, nous prendrons $\gamma = 40.1 \times 10^{-3} N.m^{-1}$, $h_0 = 180nm$ et $\lambda = 500nm$. La viscosité mesurée est ainsi de l'ordre de $10^8 Pa.s$, ce qui s'avère réaliste pour un tel matériau (PS130k) à une température de 130°.

En conclusion, ces travaux ont abouti à deux avancées. La première concerne la maîtrise du comportement des résines utilisées pour la lithographie à nano-impression. En fournissant un moyen de mesurer précisément la viscosité dynamique pour une température donnée, cette expérience permet d'optimiser les températures utilisées pendant le procédé. En maîtrisant l'évolution de la forme du créneau lors du démoulage, elle permet par exemple de déterminer la température limite pour laquelle un plus fort refroidissement est inutile. Cette viscosité a également une importance fondamentale pour le remplissage des motifs du moule.

La deuxième avancée est la mise au point du procédé de régularisation et des outils informatiques décrits dans cette thèse. Beaucoup de mesures ont été faites et un grand nombre de modèles géométriques ont été développés pour arriver finalement aux résultats présentés ici ainsi que dans le chapitre suivant.

CHAPITRE 9

Suivi de gravure plasma

L'un des objectifs visés par ces travaux de thèse est le développement d'une méthode de suivi en temps réel des procédés de **gravure plasma**. Le procédé choisi ici est la gravure de résine photosensible. C'est une étape utilisée communément par l'industrie microélectronique sous le nom de *resist trimming*. Elle sert par exemple à réduire la largeur de grille du transistor.

Lors de la réalisation de procédés dans une machine de gravure, les ions qui constituent le plasma sont accélérés et rentrent en contact avec l'échantillon. La bombardement de la matière ainsi provoqué peut entraîner :

- une éjection des couches superficielles, donc un rétrécissement de la cote du motif ;
- un changement d'état de surface si, par exemple, certains atomes de l'échantillon sont éjectés, si des espèces du plasma sont adsorbées en surface ou si le tout réagit chimiquement avec l'aide de rayonnement ultraviolet créé par le plasma. Ce phénomène entraîne la plupart du temps un changement de l'indice optique de la couche impactée [38].

Les expériences de gravure effectuées ont pour caractéristique d'être, d'une part, cohérentes avec les intérêts de la technologie microélectronique et, d'autre part, suffisamment élémentaires pour permettre un développement rapide de la scatterométrie dynamique. Il s'agira dans tous les cas présentés ici de gravure (sous différentes conditions expérimentales, avec différentes espèces) d'un simple réseau de lignes de résine utilisée pour la lithographie 248nm[1]reposant sur un substrat de silicium massif (qui se recouvre naturellement d'une fine couche d'oxyde SiO_2). Ces lignes auront une largeur et une hauteur de $500nm$ environ et une période de $1000nm$. Une représentation en est faite sur la figure 9.1.

[1]Aujourd'hui, dans l'industrie, il est plus courant d'utiliser de la résine photosensible à 193nm. Cependant nous avons choisi d'utiliser la 248 nm pour nos travaux pour plusieurs raisons :

- Cette résine est aujourd'hui largement étudiée et bénéficie d'une littérature abondante (Le lecteur pourra notamment consulter la thèse d'Erwine PARGON [78]).
- Les installations du LETI que nous utilisions (en particulier la machine de photolithographie ASML/300) nous ont permis de produire des plaques-échantillons sur place.
- Les indices optiques de la résine 248nm varient bien moins que ceux de la résine 193nm pendant le procédé de gravure.

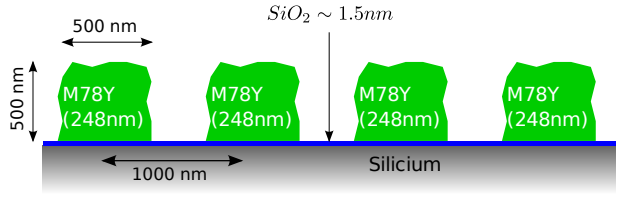


Fig. 9.1: Réseau de lignes de résine utilisé pour les expériences

Ce cas de figure présente les avantages suivants :
- Il met en jeu peu de matériaux différents : résine, silicium et oxyde de silicium (créé naturellement par oxydation du substrat) ;
- Il est simple à modéliser géométriquement : on peut se contenter de peu de paramètres pour décrire un réseau susceptible de varier quasi-homotétiquement dans le temps ;
- Les échantillons sont faciles à fabriquer : un réseau de résine sur silicium s'obtient très facilement par lithographie [56] (optique, par faisceau d'électrons ou encore par nano-impression [86])

9.1 Dispositif expérimental

Nous avons utilisé pour ces travaux un réacteur de gravure muni d'un ellipsomètre ainsi qu'un microscope à force atomique.

9.1.1 Bâti et réacteur de gravure

Le bâti de gravure disponible au LTM est une plateforme industrielle, de type *Centura 5200* construit par Applied Materials. Comme représenté sur la figure 9.2, le système est composé, pour ce qui nous intéresse :
- d'un **sas de chargement** permettant d'introduire des plaques de silicium de 200mm ;
- d'une **chambre de transfert** sous vide munie d'un bras robotisé pour le déplacement des plaques entre les chambres périphériques ;
- d'un **réacteur de gravure** plasma de type DPS (*Decoupled Plasma Source*) équipé d'un ellipsomètre Jobin-Yvon UVISEL (cf. chap. 2) pour des mesures *in situ* ;
- d'une **chambre d'orientation**. Dans notre cas, la plaque doit être tournée de 12°, de telle sorte que les lignes des réseaux soient perpendiculaires au faisceau de lumière de l'ellipsomètre.

Un réacteur plasma de type DPS permet d'obtenir des plasmas de haute densité (typiquement, de 10^{11} à 10^{12} ions par cm^3) tout en travaillant à basse pression (quelques mTorr). Tel que représenté sur le schéma 9.3, le nôtre est constitué :
- d'une **source à couplage inductif** : il s'agit d'une antenne radiofréquence entourant le haut du réacteur et constituant avec le plasma un transformateur : le courant électrique du primaire (antenne) est transformé en flux dans le plasma (secondaire). Le flux est donc ici azimutal et confine le plasma : la quantité d'espèces qui se précipite sur les parois est bien moindre que si le flux était vertical par exemple. Ceci explique pourquoi de plus faibles pressions sont permises. La fréquence habituellement utilisée, de 12.56 MHz, permet de créer et de maintenir le plasma. La puissance injectée dans cette source sert à contrôler la densité des ions.

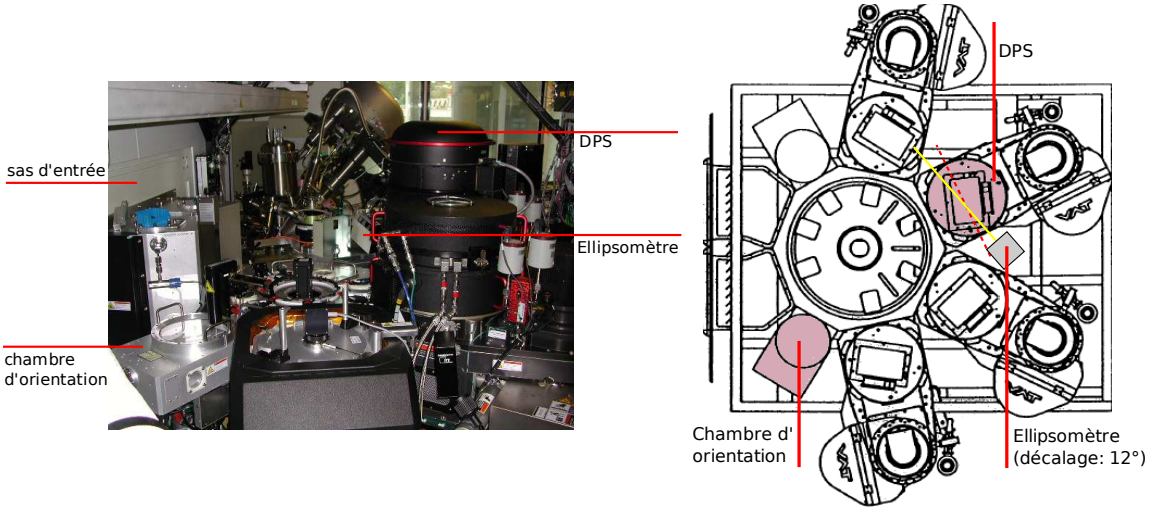


Fig. 9.2: Plateforme Centura 5200 - Applied Materials

- d'une **source à couplage capacitif** : on relie le porte-substrat (et la plaque) à un générateur radiofréquence (à 13.56 MHz, de manière à éviter les interférences, les deux générateurs étant à proximité l'un de l'autre). Pour compenser le fait que les électrons, plus mobiles que les ions, rentrent en contact plus facilement avec le substrat, celui-ci va se charger négativement. Il va ainsi provoquer son propre bombardement ionique. La puissance injectée dans cette source sert ici à contrôler l'énergie des ions et donc l'anisotropie de la gravure : la vitesse de gravure sera en effet supérieure selon la direction verticale.
- de boîtes d'accord : par un système d'asservissement, l'impédance vue par les générateurs est maintenue à 50Ω. Ceci assure une bonne adaptation d'impédance et permet au plasma d'absorber la plus grande part de l'énergie fournie, plutôt que la réfléchir.
- d'un système de pompage : Il est constitué d'une pompe turbomoléculaire de grande puissance qui permet d'évacuer jusqu'a $2000l/s$ et permet ainsi d'introduire dans le réacteur un débit de gaz allant jusqu'a $200sccm$ ([2]) tout en maintenant une pression constante. Il est constitué aussi d'une pompe primaire qui permet de faire un vide compatible avec le transfert des plaques dans la plateforme (quelques mTorr).

9.1.2 Microscope à force atomique

Un microscope à force atomique[3] est constitué d'une sonde de très petite taille (quelques dizaines de nanomètres) qui, fixée à un levier, vient interagir avec la surface d'un échantillon à analyser. Cette interaction de faible distance, est double : attractive par de la force de Van der Waals et répulsive à cause des nuages d'électrons entourant les atomes qui se repoussent. C'est donc une distance d'équilibre entre la sonde et l'échantillon qui sera mesurée.

Deux modes de mesure sont habituellement utilisés :

- Le **mode contact** : la pointe vient toucher la surface et on mesure la déviation du levier. Cette mesure se fait à l'aide d'un rayon laser que l'on fait réfléchir sur le levier et dont on mesure la déviation (cf. schéma 9.4(a)) ;

[2] *standard centimeter cube per minute*, centimètres cube de gaz dans les conditions standard par minute

[3] Il sera désigné par la suite par **AFM**, sa forme anglaise habituellement utilisée : *atomic force microscope*

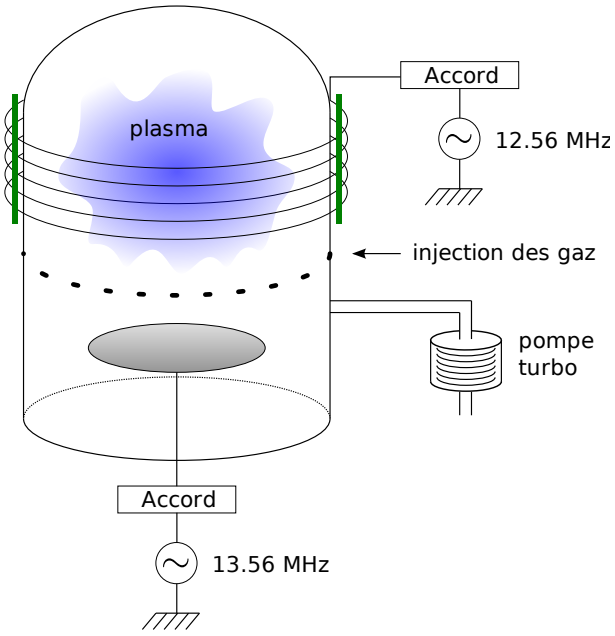


Fig. 9.3: Principe de fonctionnement du réacteur plasma à sources découplées

– Le **mode tapping** : on fait vibrer la pointe à sa propre fréquence de résonance. On mesure alors l'amplitude de la vibration ; elle décroît en effet à l'approche d'une surface. Cette technique est bien plus utilisée car elle génère moins d'usure pour la pointe, le contact se faisant par intermittence (cf. schéma 9.4(b)).

L'intérêt d'une mesure AFM est sa précision de l'ordre du nanomètre. On se servira donc de cet outil pour évaluer l'autre technique de métrologie qu'est la scatterométrie, ceci en gardant à l'esprit que la forme de la pointe entre en jeu dans la mesure des motifs. Ainsi, certains endroits comme les pieds d'un créneau de résine ne sont pas mesurables si la courbure est inférieure à celle de la pointe. Comme le montre le schéma 9.5, le rayon de courbure minimal de la sonde (situé sur les cotés) est de l'ordre de 20 à 30nm.

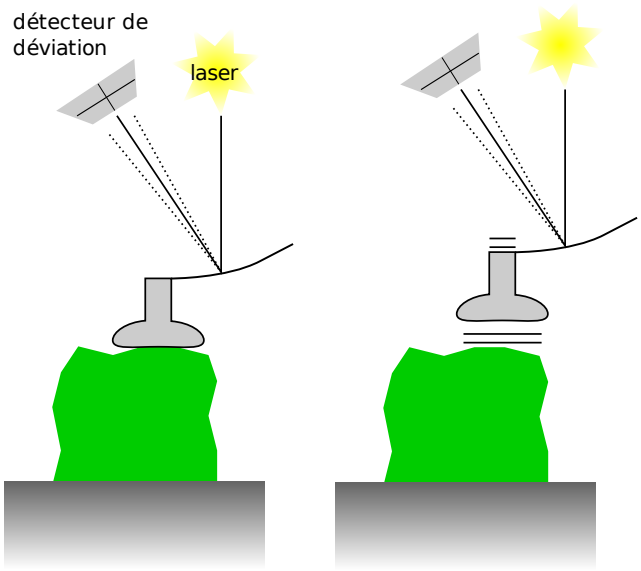
D'autre part, la longueur de la pointe doit être choisie en fonction de la hauteur des motifs, sous peine de ne pas pouvoir mesurer les zones les plus profondes. Nous avons choisi une pointe de 600nm ; elle correspond aux motifs que l'on utilisera, ceux-ci étant d'une hauteur maximale d'environ 500nm.

Pour nos mesures, nous disposions d'un AFM 3D capable de mesurer des motifs verticaux en utilisant cette forme de sonde en patte d'éléphant ; avec ce type d'appareil, nous avions ainsi accès à la variation de la largeur du motif sur toute sa hauteur. Ainsi, afin de s'affranchir le plus possible de la rugosité présente sur les flancs, les mesures AFM utilisées sont des moyennes sur une vingtaine de profils de ligne successifs (cf. fig. 9.6).

9.2 Résultats

Les expériences de suivi de procédés de gravure plasma ont toutes été faites sur des échantillons identiques décrits dans l'introduction de ce chapitre. Seules les conditions expérimentales du plasma ont varié ; elles sont déterminées ici par :

– Les espèces en jeu dans le plasma et leur débit d'introduction dans le réacteur ;



(a) mode *contact* (b) mode *tapping*

Fig. 9.4: Modes de fonctionnement du microscope à force atomique

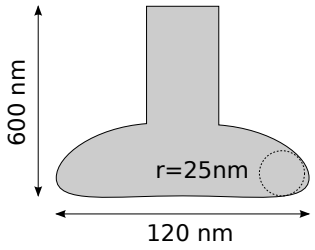


Fig. 9.5: Forme de la sonde AFM utilisée

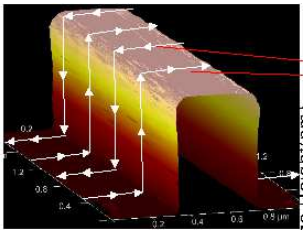


Fig. 9.6: Mesure de plusieurs profils successifs pour annuler la rugosité de flanc.

– Les puissances des sources inductive et capacitive ;
– La pression du plasma.

Plusieurs fois dans la suite, on remarquera l'état inachevé de certaines expériences. Cela est dû à plusieurs pannes intervenues à la fin de cette thèse sur la plateforme de gravure Centura : casse du bras motorisé, puis problème de pressurisation.

9.2.1 Gravure par plasma $HBr - O_2$

Les conditions expérimentales sont ici : plasma $HBr - O_2$ introduit à $70sccm/30sccm$, pression maintenue à $4mTorr$, puissance source (inductive) de $300W$, puissance *bias* (capacitive) nulle.

Cette expérience a consisté à graver en plusieurs fois une plaque munie du réseau. Chaque étape est suivie par l'ellipsomètre pour acquérir le "film" des signatures scatterométriques. Au début de la gravure et à la fin de chaque étape, une mesure AFM est effectuée pour la comparaison avec la scatterométrie. Cette expérience nécessite donc beaucoup de manipulations : après l'établissement du plasma et le temps de gravure, il faut retirer la plaque du bâti de gravure, nettoyer les parois du réacteur[4], porter la plaque dans le microscope à force atomique, puis revenir ensuite à l'étape de gravure.

Comme nous allons le voir, les premiers résultats nous ont enseigné que cette manière de faire pose un problème de nature expérimentale. En effet, à la fin de chaque étape, le réacteur se trouve nettoyé ; or cela change la nature du plasma et influe sur la vitesse de gravure de l'étape suivante.

La durée totale de la gravure peut être connue à l'avance en gravant une plaque témoin et en notant le temps au bout duquel la réponse ellipsométrique varie brutalement. Cette variation se produit quand la résine a complètement disparu de la plaque, et cette technique s'appelle une **détection de fin d'attaque**. Cette durée, ici d'approximativement 500 secondes, a été divisée en étapes. Trois fois 80 secondes d'abord, puis 2 fois 40 secondes, comme cela est schématisé sur la figure 9.7. Cela ne nous amène qu'à $t = 320s$, car l'expérience est inachevée.

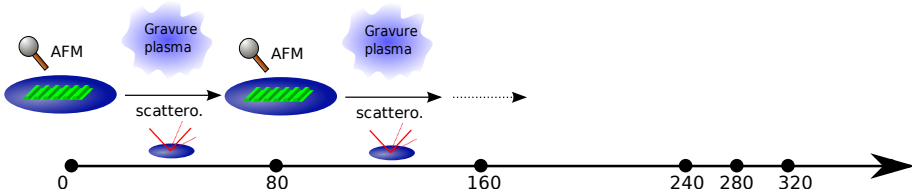


Fig. 9.7: Expérience de gravure d'un réseau de ligne par plasma $HBr - O_2$

Les mesures AFM faites à chaque étape sont représentées sur le graphique 9.8. Elles permettent d'élaborer un modèle géométrique pour le suivi par scatterométrie. On a choisi le modèle le plus simple, paramétré par une largeur de créneau (CD) et une hauteur (H).

On calcule une bibliothèque dont les paramètres sont donnés dans le tableau 9.1.

	min	pas	max
CD	0	1	500
H	0	2	580
période	1000		

Tab. 9.1: Bibliothèque de signatures scatterométriques

La variation de ces paramètres lors de la gravure plasma est représentée sur la figure 9.9. Sur cette figure se trouvent tracés les paramètres H et CD variant au cours du temps (en lisse, la variation

[4]La plupart des procédés de gravure entraînent un dépôt de matériaux sur la paroi du réacteur. Ce dépôt conditionne fortement la gravure et il convient donc, pour ne pas perturber les autres expériences qui utilisent la machine, de nettoyer ces parois. Cela se fait au moyen d'un plasma $SF_6 - O_2$.

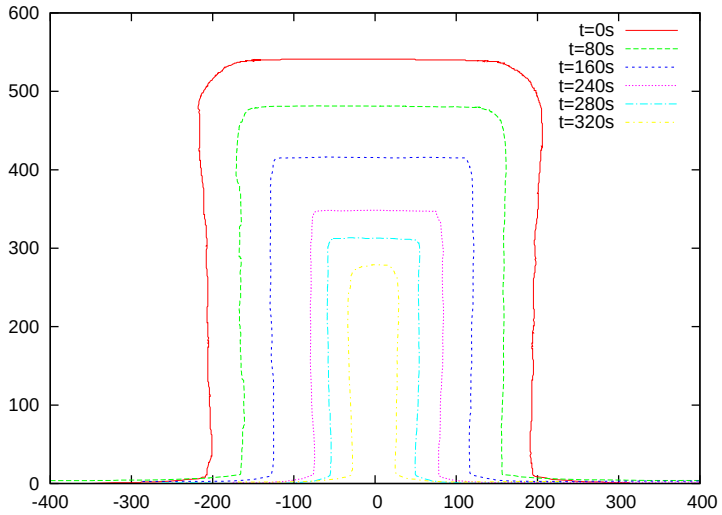


Fig. 9.8: Évolution du profil d'un réseau gravé par plasma $HBr - O_2$ observé par AFM. Les grandeurs sont en nanomètres et représentent une moyenne de 20 profils mesurés.

régularisée, en croix, le résultat à chaque instant de la recherche d'un seul PPV dans la bibliothèque) ainsi que des points de comparaison avec la microscopie à force atomique. Le graphique du bas est un rapport du processus de reconstruction :

- les croix rouges représentent à chaque instant lequel des plus proches voisins a été choisi (ici, nous recherchons 16 P.P.V.). Une croix en bas signifie que le 1^{er} plus proche voisin a été choisi, alors qu'une croix en haut indique que c'est le 16^{e}. Dans ce cas particulier, la répartition des croix est homogène ce qui signifie que l'algorithme de régularisation a agi. Dans le cas contraire, si la répartition est inhomogène, cela signifie que les données de base sont déjà trop régulières ou que les paramètres de la pré-régularisation β_1 sont trop faibles, c'est-à-dire que l'algorithme a eu trop peu de latitude pour reconstruire la régularité.
- la courbe continue rouge représente le résidu issu de la pré-régularisation (différence entre signature acquise et signature choisie parmi les 16) : la variation de ce résidu donne une estimation (qualitative) de la "quantité de proximité qui a été compensée par de la régularité".

On donne dans le tableau 9.2 les différences (en nm) entre les dimensions relevées par AFM et mesurées par scatterométrie.

En utilisant le même modèle géométrique, nous avons analysé le film de signatures issu de la gravure de la plaque témoin. Les variations de la largeur et de la hauteur du créneau sont représentées sur la figure 9.10.

Les variations du profil pour l'expérience témoin sont visiblement différentes de celles de l'-expérience interrompue plusieurs fois ; les motifs disparaissent au bout de 470s pour la première et 320s pour la dernière. Cela traduit une vitesse de gravure de 30% plus grande environ lorsque (pour des conditions identiques par ailleurs) la paroi du réacteur est nettoyée plusieurs fois. Ceci s'explique par le fait que la concentration d'oxygène atomique - espèce qui est responsable de l'essentiel de la gravure - diminue si la paroi est tapissée d'un dépôt. Dans ce cas, l'espèce O s'adsorbe ou participe à une réaction physico-chimique générant un gaz qui chasse l'oxygène du réacteur.

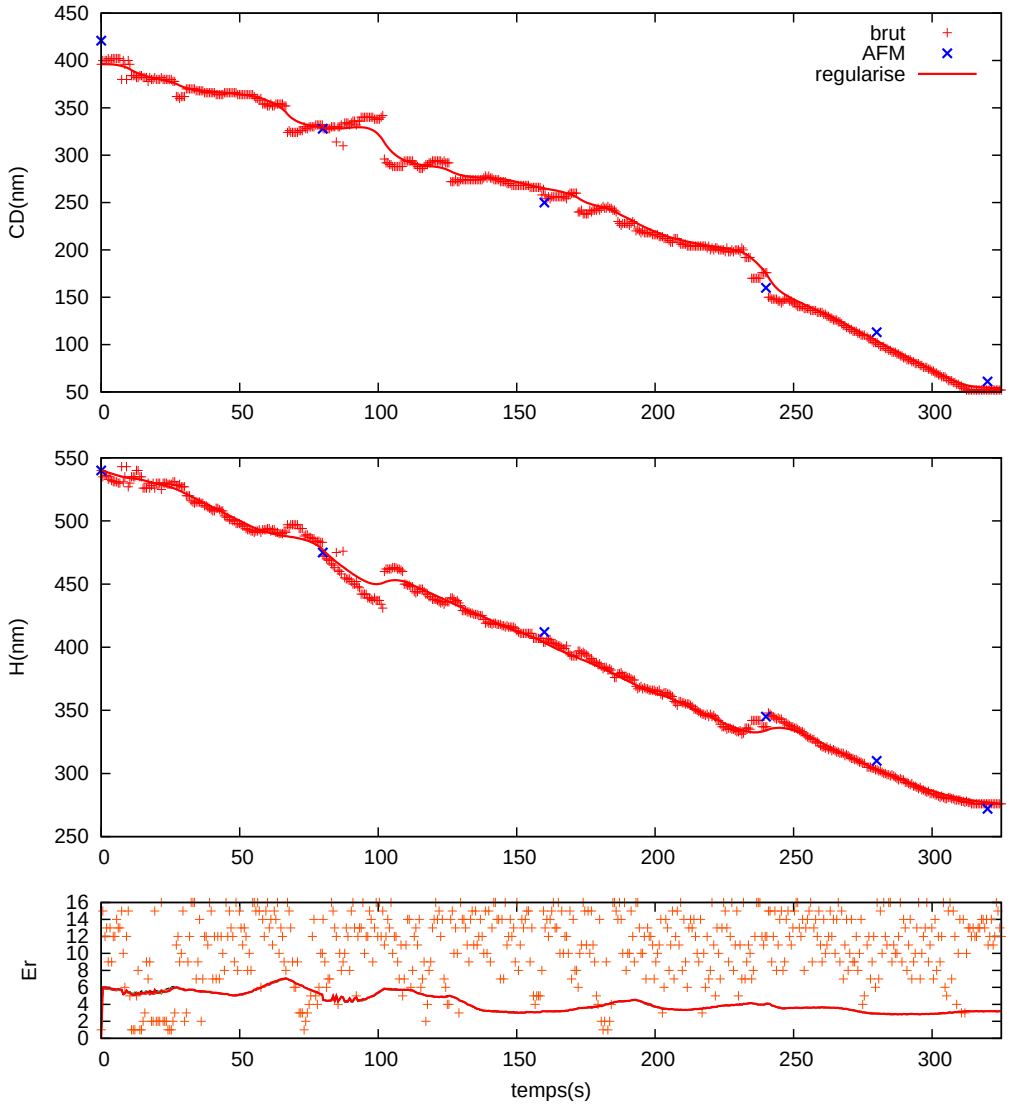


Fig. 9.9: Évolution des paramètres géométriques du profil d'un réseau gravé par plasma $HBr - O_2$

t(s)	CD(nm) AFM	scattero.	Δ	H(nm) AFM	scattero.	Δ
0	421	421	-	540	540	-
80	328	328.9	0.9	475	476.6	1.6
160	250	264,8	14.8	412	403.3	8.7
240	160	173.7	13.7	345	334.3	10.7
280	113	103.8	9.2	310	303.8	6.2
320	61	54.4	6.6	272	277.1	5.1

Tab. 9.2: Comparaison quantitative entre AFM et scatterométrie

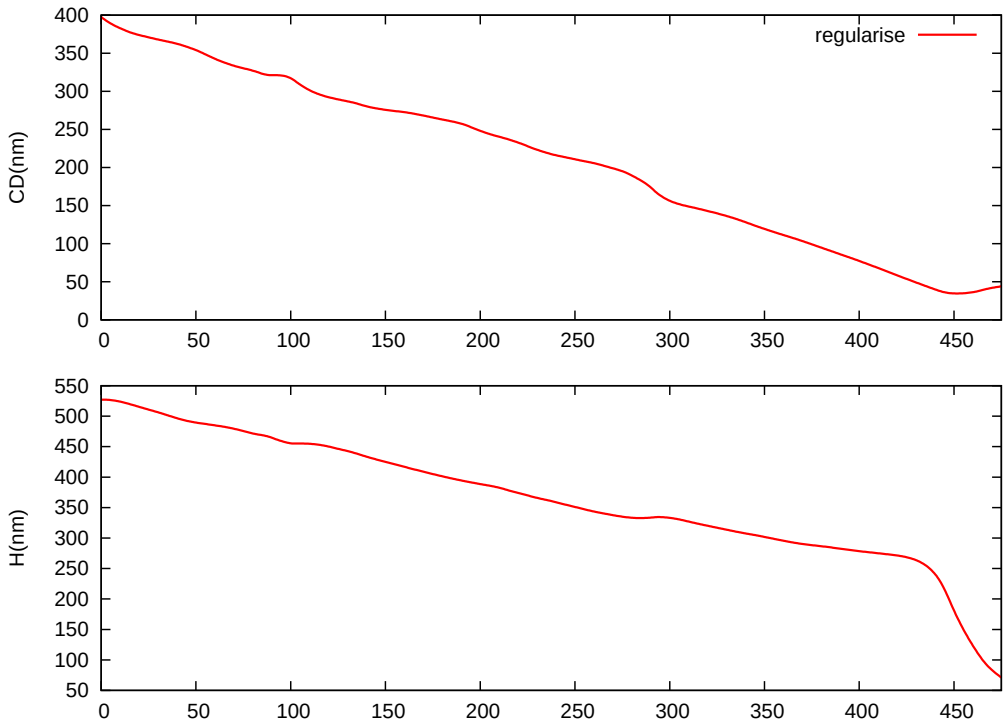


Fig. 9.10: Évolution des paramètres géométriques du réseau de la plaque témoin.

9.2.2 Gravure par plasma $Ar - O_2$

Nous souhaitons comparer ici l'influence des espèces qui constituent le plasma. Les conditions sont identiques à la gravure $HBr - O_2$, mais l'argon est utilisé à la place du bromure d'hydrogène.

Instruits par l'expérience de gravure $HBr - O_2$ des problèmes de déconditionnement de parois, nous avons pour cette nouvelle expérience changé la manière d'opérer. Ici, une plaque sur laquelle a été étalée de la résine (une simple couche, sans motifs imprimés) est consacrée à rétablir les conditions de la paroi qui ont été changées lors de l'extraction de la plaque du réacteur.

En pratique, à chaque étape, avant de réintroduire dans le réacteur une plaque (avec motif) déjà gravée pendant T secondes, on procède à une gravure dans les mêmes conditions et pendant T secondes de la plaque *sacrificielle*. L'étape de substitution dans le réacteur de la plaque plane par la plaque avec motif se fait manuellement et sans délais. Ce principe est schématisé sur la figure 9.11.

Les résultats, pour le modèle simple (créneau carré de résine sur silicium), sont représentés sur la figure 9.12. Le "film" de mesures scatterométriques est complet ; il correspond en effet à la gravure *en une fois* de la plaque témoin (qui sert à déterminer le temps de gravure total). En revanche, seules quatre mesures AFM sont représentées : la poursuite de cette expérience devrait fournir le reste.

On donne dans le tableau 9.3 les différences (en nm) entre les dimensions relevées par AFM et mesurées par scatterométrie.

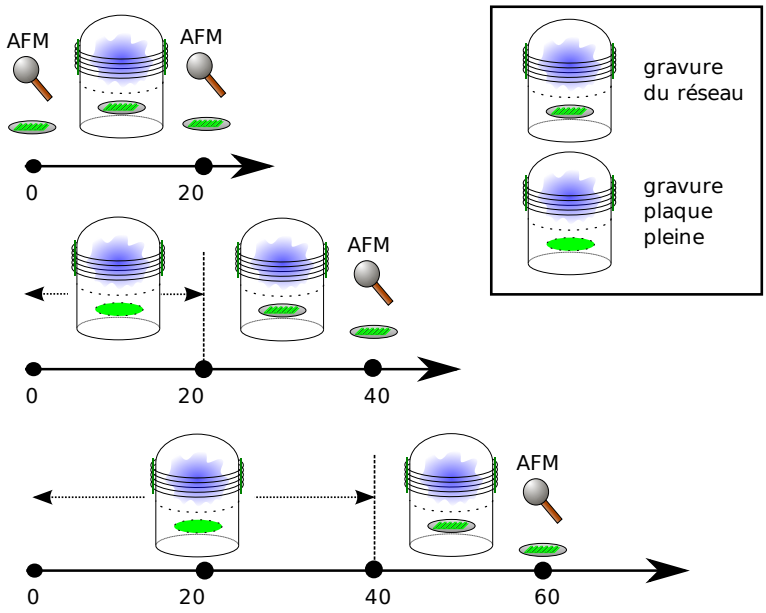


Fig. 9.11: Expérience de gravure d'un réseau de ligne par plasma $Ar - O_2$

	CD(nm)			H(nm)		
t(s)	AFM	scattero.	Δ	AFM	scattero.	Δ
0	402.3	402.3	-	550.3	550.3	-
22	338.7	340.8	2.1	496.6	491.2	5.4
42	285.9	296.3	10.4	451	446.7	4.3
62	249.7	257.8	8.1	400	400.34	0.34

Tab. 9.3: Comparaison quantitative entre AFM et scatterométrie

Ces premiers résultats de gravure mettent en lumière un certain nombre d'enjeux pour le développement d'une méthode de métrologie précise (à la lecture des tableaux 9.2 et 9.3, nous sommes loin ici des objectifs de l'ITRS) pour le suivi de gravure *in situ* et en temps réel. Ces enjeux sont liés notamment à la modélisation du réseau diffractant au sens large (puisque la scatterométrie en tant que telle est déjà validée pour de grandes précisions) :

- Quels indices optiques utiliser ? Les réactions physico-chimiques à l'intérieur du réacteur ainsi que le rayonnement ultra-violet du plasma peuvent changer, en surface au moins, la nature des matériaux. Cet aspect des choses fait l'objet de beaucoup de recherches et la scatterométrie dynamique devra s'appuyer sur ces résultats de physique des plasmas de manière probablement très poussée.
- Quels modèles géométriques ? Nous n'avons, pour l'exemple, utilisé que deux paramètres (hauteur et largeur). Or la gravure plasma d'un système comportant plusieurs matériaux engendre, à cause de sa sélectivité, des profils aux géométries complexes : arrondis aux pieds et à la tête du créneau, angles et facettes multiples, etc. Là encore, la clé est dans la maîtrise du procédé, et donc dans l'information introduite *a priori* dans le problème inverse. Dans ce cas particulier, il s'agit d'une modélisation géométrique rigoureuse.

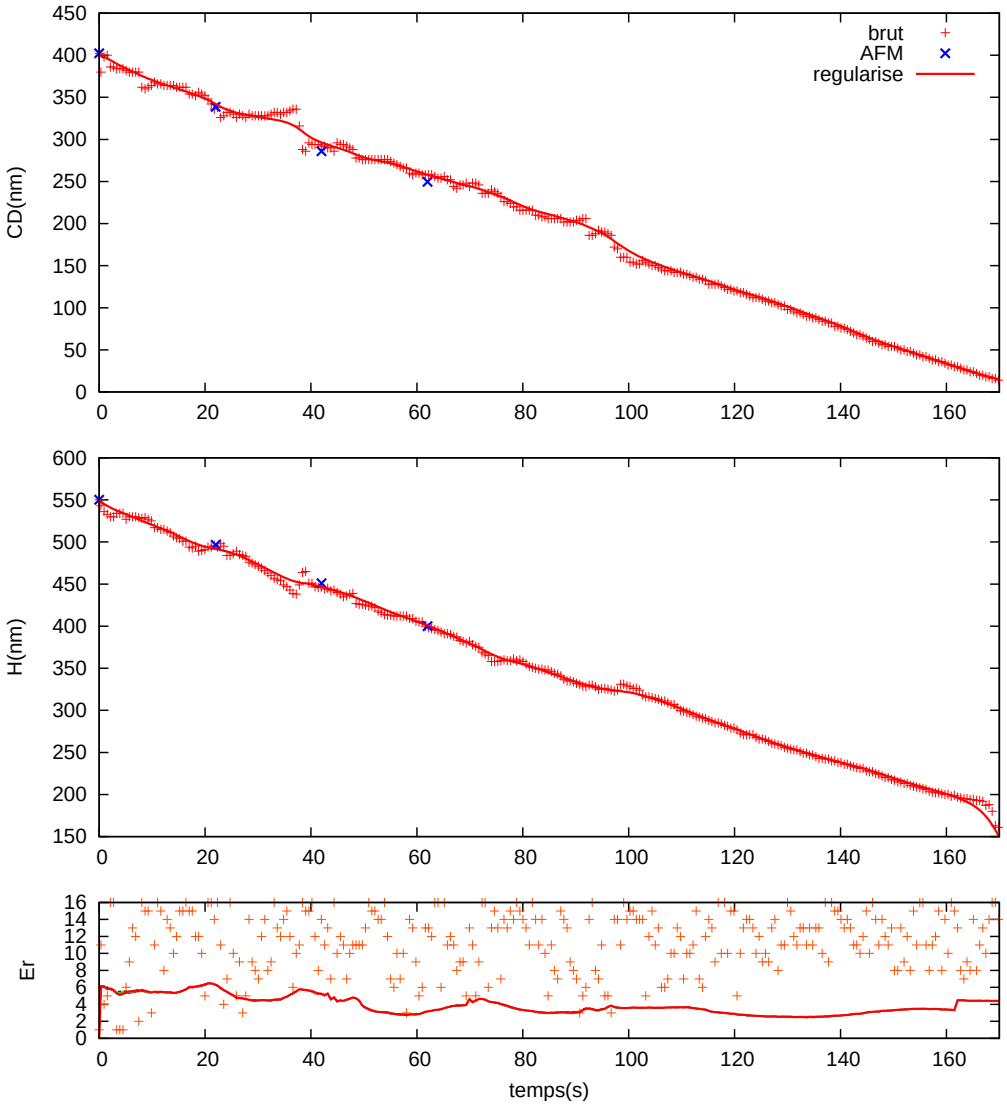


Fig. 9.12: Évolution des paramètres géométriques du profil d'un réseau gravé par plasma $Ar - O_2$

Cette thèse s'inscrit donc dans les prémisses du développement d'un système de métrologie *in situ* et temps réel pour la gravure plasma. Une autre thèse est en cours au laboratoire ; elle devrait permettre de faire avancer ces problématiques majeures.

Quatrième partie

Détermination des indices optiques des matériaux

Un des points fondamentaux pour une scatterométrie efficace est la connaissance précise des propriétés optiques des matériaux constituant le motif diffractant. Une signature scatterométrique est en effet autant dépendante des indices optiques de ces matériaux [37] que de la géométrie du profil diffractant.

La mesure de l'indice optique d'un matériau semi-infini peut se faire directement par ellipsométrie : Connaissant l'angle d'incidence θ_0 de la lumière sur le matériau, l'indice $\tilde{n}_0$ du milieu ambiant ainsi que la réponse de l'ellipsomètre à l'excitation lumineuse ρ, on démontre, en utilisant les coefficients de Fresnel (3.2 et 3.3) et la relation de Snell-Descartes (3.1), que :

$$\tilde{n}_1 = \tilde{n}_0 \sin\theta_0 \left[1 + \frac{1-\rho^2}{1+\rho}\tan^2\theta_0\right] = \tan^2\theta_0 \left[1 - \frac{4\rho}{(1+\rho)^2}\sin^2\theta_0\right] \tag{9.1}$$

Néanmoins, comme il est rarement possible d'obtenir des échantillons suffisamment massifs pour ne pas observer de réflexion sur un éventuel substrat, la détermination des valeurs n et k d'un matériau se fait le plus souvent par réflectométrie ou ellipsométrie spectroscopique sur couche mince (cf. le chapitre 2 traitant de l'ellipsométrie). Le procédé consiste à coucher sur un substrat d'indices parfaitement connus (nous utilisons des plaques de silicium dédiées à la microélectronique) une couche mince du matériau à déterminer, et à mesurer ensuite les grandeurs optiques significatives, longueur d'onde par longueur d'onde.

En regard des équipements auxquels nous avons eu accès pendant ces travaux de thèse, la méthode utilisée dans la suite sera basée sur des mesures ellipsométriques spectroscopiques. Le principe de détermination d'indices exposé est néanmoins parfaitement applicable à des mesures réflectométriques.

Notre problème est un **problème inverse** : s'il est aisé, avec les équations de Fresnel, de calculer une signature ellipsométrique spectroscopique connaissant les indices et l'épaisseur de la couche, le problème réciproque est généralement plus difficile à résoudre correctement :

$$(n, k) \xrightarrow{f^{-1}} (e, D_1, D_2)$$

(Où D_1 et D_2 sont deux signaux ellipsométriques quelconques (I_s/I_c, S_1/S_2, etc.) et e l'épaisseur de la couche du matériau.)

CHAPITRE 10

Modèles de lois de dispersion

La variations des valeurs de n et k peuvent être modélisées par des **lois de dispersion**. Ce sont des fonctions de la variable λ (ou de l'énergie $h\nu$ indifféremment) qui dépendent de relativement peu de paramètres ; déterminer ces paramètres permet de définir ces fonctions sur tout le spectre. Dans la suite, plusieurs modèles de lois de dispersion seront présentés et leur validité évaluée.

Un critère important de validité est notamment la vérification des relations de Kramers-Krönig ([47], [46]). Ces relations sont un lien entre la partie réelle $n(\omega)$ et la partie imaginaire $k(\omega)$ (ω étant la pulsation de l'onde, reliée à la longueur d'onde par $\omega = 2\pi c/\lambda$) de l'indice optique complexe. Elles expriment le principe de causalité : un matériau ne peut être excité électriquement avant l'existence de l'onde incidente.

$$\begin{cases} n(\omega) = 1 + \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{k(\omega')}{\omega' - \omega} d\omega' & (10.1) \\ k(\omega) = -\frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{n(\omega')}{\omega' - \omega} d\omega' & (10.2) \end{cases}$$

On remarque que ces intégrales sont à calculer sur tout $\mathbb{R}$, ce qui rend difficile en pratique l'usage de cette relation.

10.1 Modèles empiriques

Les modèles présentés ici ont pour seule ambition de générer des courbes de dispersion ressemblant le plus possible à la courbe réelle sur au moins une partie du spectre. Ce sont des modèles simples mais à la validité toute relative ; ils ne respectent notamment pas les relations de Kramers-Krönig.

10.1.1 Modèle de Cauchy

Ce modèle, élaboré en 1836 par le français Louis Augustin Cauchy, consiste à exprimer les indices optiques du matériau de cette manière :

$$\begin{cases} n(\lambda) = \sum_{i\geq 1} \frac{n_i}{\lambda^{2(i-1)}} \\ k(\lambda) = \sum_{i\geq 1} \frac{k_i}{\lambda^{2(i-1)}} \end{cases}$$

La plupart du temps, on n'utilise que les deux premiers termes de la somme ; seulement 4 paramètres sont alors nécessaires (n_1, n_2, k_1 et k_2).

Cette description est rarement valable car son comportement monotone ne permet pas de modéliser des pics de dispersion, comme le montre le graphique 10.1. Cependant, on s'en sert couramment pour des matériaux que l'on sait transparents ($k = 0$) sur une plage de longueur d'onde donnée (en général le visible : 400nm-800nm).

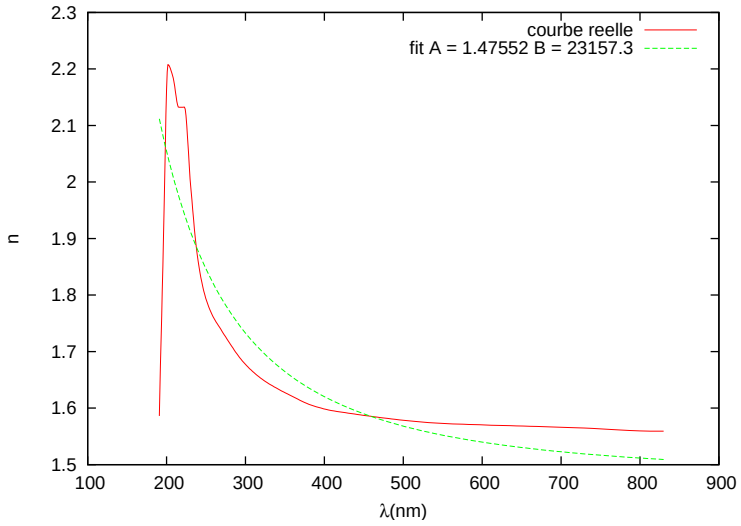


Fig. 10.1: Modèle de Cauchy appliqué à une couche de 134,2nm de polystyrène de masse molaire 28kg/mol. Le résultat de la régression donne : $n_1 = 1,47$ et $n_2 = 23157$.

10.1.2 Équation de Sellmeier

Ce modèle dédié aux matériaux transparents ($k = 0$) fut développé vers 1871 par Sellmeier à la suite des travaux de Cauchy dans le but de pallier l'impossibilité de modéliser les pics de dispersion :

$$n^2(\lambda) = 1 + \sum_{i\geq 1} \frac{B_i \lambda^2}{\lambda^2 - C_i} \tag{10.3}$$

En général, on utilise les trois premiers termes de la somme. Une représentation de ce modèle est donnée sur la figure 10.2.

Le problème de ce modèle est que les pics sont modélisés par des limites de n^2 en l'infini quand $\lambda \to C_i$; cela n'a rien de physique.

10.1.3 Modèle polynomial

Ce modèle sert le plus souvent à modéliser le comportement de certains métaux conducteurs dans le visible.

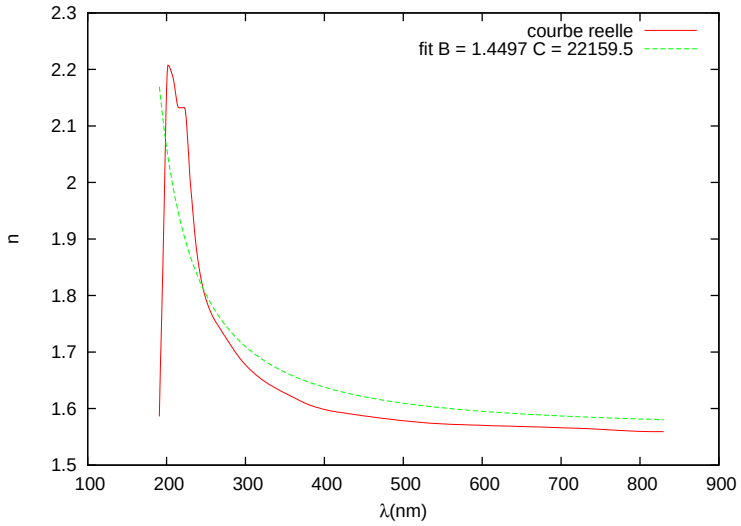


Fig. 10.2: Modèle de Sellmeier appliqué à une couche de 134,2nm de polystyrène de masse molaire 28kg/mol. Le résultat de la régression avec un seul terme donne : $B_1 = 1,44$ et $C_1 = 22159$.

$$\begin{cases} n(\lambda) = \sum_{i \geq 1} n_i \lambda^{2(i-1)} \\ k(\lambda) = \sum_{i \geq 1} k_i \lambda^{2(i-1)} \end{cases}$$

Pour coller au comportement parabolique dans le visible, ces métaux doivent avoir un pic d'absorption décalé vers l'infrarouge. Dans ces travaux de thèse, nous n'avons utilisé que le silicium. Or celui-ci est un semi-conducteur qui possède un pic d'absorption dans l'ultraviolet (cf. fig. 10.3). Donc dans notre cas, le modèle n'est pas utilisable.

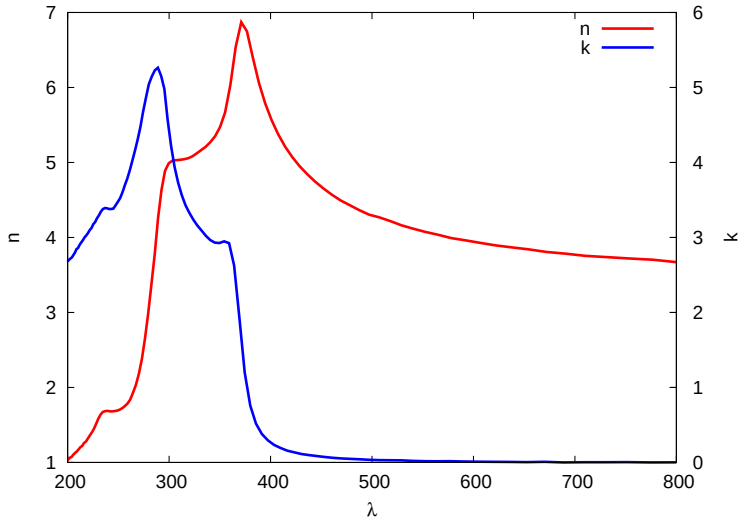


Fig. 10.3: Courbes de dispersion du silicium utilisé en microélectronique

10.2 Modèles avec oscillateurs

Ce modèle s'appuie cette fois véritablement sur des considérations physiques : on considère que la permittivité diélectrique d'un matériau (donc, ses indices optiques[1]) est liée à l'action des électrons qui absorbent le rayonnement et le restituent de manière altérée. Ces modèles respectent les relations de Kramers-Krönig.

Le modèle de l'**oscillateur de Lorentz** consiste à modéliser l'interaction lumière-matière par un électron élastiquement lié à un atome (ou une molécule), excité par le champ électrique harmonique et subissant une force de résistance proportionnelle à sa vitesse. Une combinaison d'équations simples de mécanique classique et d'électromagnétisme (voir [76] p.219) permet alors d'exprimer la permittivité diélectrique du matériau par :

$$\tilde{\epsilon} = \tilde{\epsilon}_\infty + \frac{F}{-\omega^2 + i\gamma\omega + \omega_0^2}$$

où $\tilde{\epsilon}_\infty$ est la permittivité du fond ($\tilde{\epsilon}_\infty = 1$ pour le vide), γ le coefficient de la force de résistance et ω_0^2 la constante de raideur du ressort. ω apparaît donc comme une fréquence de résonance et c'est là l'intérêt du modèle : pouvoir décrire un pic dans la courbe d'absorption du matériau.

La mécanique quantique permet d'étendre ce modèle en considérant, pour le mouvement de l'atome, plusieurs modes propres d'intensité différentes f_k et donc plusieurs fréquences de résonance ω_k et plusieurs coefficients d'atténuation γ_k :

$$\tilde{\epsilon} = \tilde{\epsilon}_\infty + \sum_k \frac{f_k\omega_k^2}{-\omega^2 + i\gamma_k\omega + \omega_0^2}$$

L'avantage ici est de pouvoir modéliser autant d'oscillateurs harmoniques qu'il y a de pics dans la courbe d'absorption, au prix naturellement de la détermination de 3 paramètres par modes de vibration (ω_k, f_k et γ_k) plus un, éventuellement : $\tilde{\epsilon}_\infty$.

Les quelques modèles empiriques ou basés sur des oscillateurs présentés ici font partie des plus utilisés pour la détermination des indices optiques ; ils s'appliquent en effet très bien à certaines catégories de matériaux présentant peu de variations (peu de pics notamment). En revanche, lors de la caractérisation de résines spéciales par exemple, dont la formule chimique, particulièrement élaborée, génère des variations plus complexes, ces lois sont largement en échec. Il serait donc très intéressant de disposer d'une autre technique de détermination d'indices, qui ne se base cette fois sur aucune loi. C'est l'objet du chapitre suivant.

[1] On rappelle que $\tilde{n} = n - ik = \sqrt{\tilde{\epsilon}}$

CHAPITRE 11

Reconstruction par bibliothèques

En réutilisant les principes et les algorithmes mis en oeuvre pour la reconstruction dynamique de paramètres géométriques, nous avons élaboré une technique originale pour déterminer les indices optiques d'un matériau (cf. 1.2) à partir d'une signature obtenue par ellipsométrie sur couche plane (chapitre 2).

Cette méthode ne s'appuie sur aucune loi de dispersion telles que celles présentées dans le chapitre précédent. Elle vise à caractériser des matériaux dont l'indice optique complexe est quelconque et mal connu.

11.1 Méthode des bibliothèques

Puisqu'il s'agit, dans ce cas aussi, d'un problème inverse, notre méthode consiste à faire les analogies suivantes avec la scatterométrie :

	scatterométrie	**détermination d'indices**	
temps	t	$h\nu$	énergie d'un photon
géométrie	$\mathbf{p}$	$\mathbf{n} = (n, k)$	indices optiques
signature	$\mathbf{s}$	$\mathbf{D} = (D_1, D_2)$	signaux ellipsométriques
problème direct	$\mathbf{s} = MMFE(\mathbf{p})$	$\mathbf{D} = Fresnel(\mathbf{n})$	
bibliothèque	$\mathcal{B} = \{\mathbf{p}, \mathbf{s}\}$	$\mathcal{B}_{h\nu} = \{\mathbf{n}, \mathbf{D}\}$	une bibliothèque par énergie

La manière d'opérer la reconstruction de paramètres géométriques au cours du temps sera ici appliquée à la reconstruction des variations de (n, k) selon l'énergie $h\nu$. Une différence cependant : alors que la bibliothèque utilisée à chaque itération est toujours la même dans le cas de la scatterométrie, elle sera renouvelée à chaque énergie pour la détermination d'indices ; ceci parce que l'énergie $h\nu$ est une des données du calcul de la bibliothèque $\mathcal{B}_{h\nu}$.

Pour illustrer le procédé, nous prendrons l'exemple d'une couche mince de polystyrène de masse molaire $28 kg.mol^{-1}$ (nommée par la suite *poly28k*) disposée sur substrat de silicium et chauffée à 140°. Une signature ellipsométrique produite avec un angle d'incidence de 70° est représentée figure 11.1.

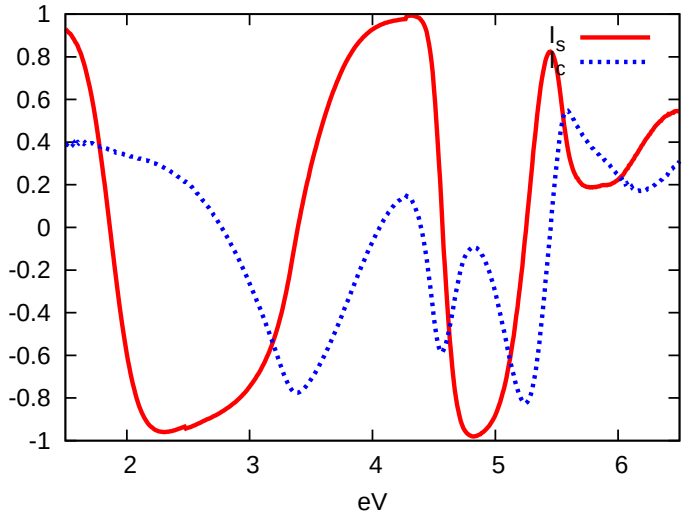


Fig. 11.1: Exemple de signature ellipsométrique : couche de 134,2nm de résine *poly28K* chauffée à 140°

On estime l'épaisseur de la couche de matériau par une méthode déjà connue et que l'on sait valable sur au moins une partie du spectre (le modèle de Cauchy transparent par exemple, cf. chapitre 10 s'applique à beaucoup de polymères dans la gamme du visible). Cette épaisseur, e, servira à créer une bibliothèque pour chaque énergie $h\nu$:

$$\mathcal{B}_{h\nu} = \{\mathbf{n}, \mathbf{D}\} = \{(n,k),(D_1,D_2)\}$$

en faisant varier les valeurs du coefficient d'absorption n et d'extinction k.

Pour chaque énergie, les k-P.P.V.du couple (D_1, D_2) acquis par l'ellipsomètre seront identifiés dans la bibliothèque. Sur la figure 11.2 on a représenté, **en projection sur les plans** $(h\nu, n)$ **et** $(h\nu, k)$ les k couples (n, k) extraits de la bibliothèques.

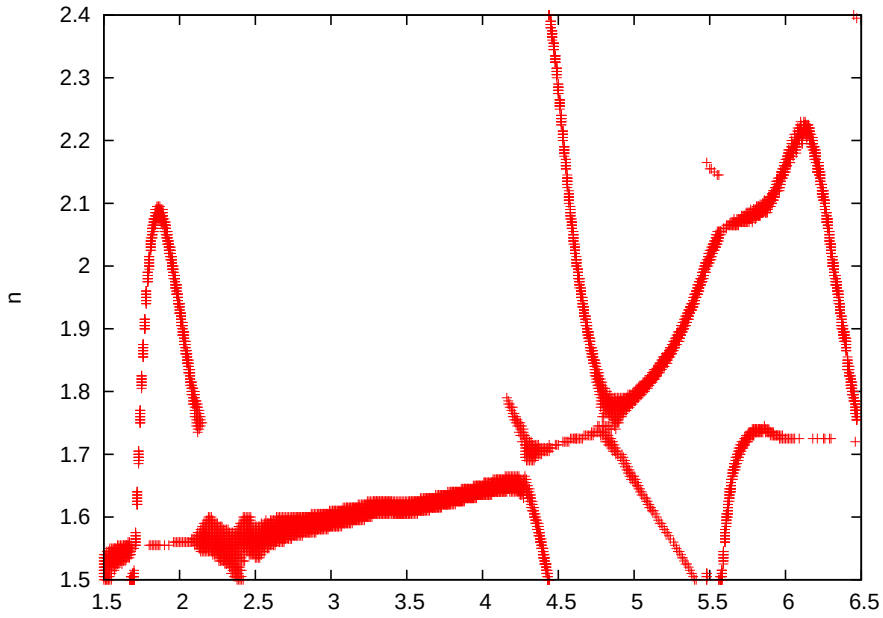
La solution à extraire fait partie des "traces" rouges représentées sur le graphique.

Notre méthode à l'avantage d'être très visuelle ; elle explique très bien par exemple les aberrations issues de certaines autres techniques : il est en effet facile de faire converger un algorithme dans un minimum local qui ne correspond pas à une solution physique mais dans lequel pourtant une fonction coût est minimale.

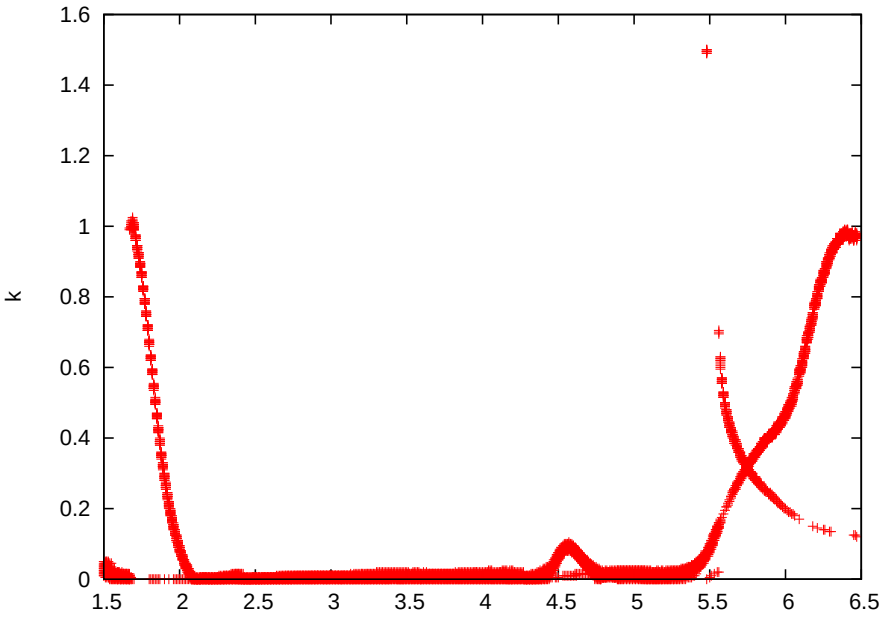
L'intérêt de représenter ainsi les solutions potentielles pour (n, k) est donc la possibilité pour l'utilisateur de "gommer" véritablement les traces inadéquates. Avec l'expérience qu'il a du matériau (le polystyrène est un polymère bien connu), l'utilisateur saura sélectionner grossièrement les traces qui peuvent être raisonnablement considérées comme faisant partie de la solution. Dans le cas où le matériau serait inconnu, certaines règles peuvent aider :

- S'agit-il d'un métal ? Dans le cas contraire, les traces à très fort coefficient d'extinction (k) seront à gommer.
- En dépit d'éventuels pics d'absorptions (valeurs localement élevées de l'indice), on considère que les grandeurs sont continues. On peut donc écarter toute trace divergente ou ne participant pas à une variation continue.

De manière analogue à la reconstruction de paramètres géométriques pour la scatterométrie dynamique, l'étape du choix du nombre de k-P.P.V. est intimement lié à la densité de la bibliothèque.



(a) coefficient d'absorption n



(b) coefficient d'extinction k

Fig. 11.2: Indices optiques des k éléments choisis de la bibliothèque (ici, $k = 16$). En abscisse est représentée l'énergie $h\nu$ du rayonnement.

En effet, si la bibliothèque est trop dense (lors de cette étape, du moins), les premiers (voire tous) k-P.P.V. seront localisés au même endroit, dans l'extremum global (ils en auront la place). Il n'y aurait pas d'alternative dans le choix de minimums locaux, c'est-à-dire pas de traces à écarter. Une augmentation du nombre k de plus proches voisins compenserait cela (puisque les derniers P.P.V. seraient susceptibles, faute de place, d'aller peupler d'autres minimums locaux), mais s'avère inutile dans cette étape qui consiste simplement à sélectionner grossièrement la solution.

Pour l'exemple, nous avons choisi de sélectionner la solution en la contraignant dans un canal. Sur la figure 11.3, on a représenté un tel canal en pointillés bleus pour la grandeur n. Un tel canal est ici déterminé à l'aide d'une courbe de dispersion connue et similaire (même matériau mais à température ambiante) : on décale la courbe de ± 0.2 pour n comme pour k. Mais l'idéal serait de dessiner ce canal à la main.

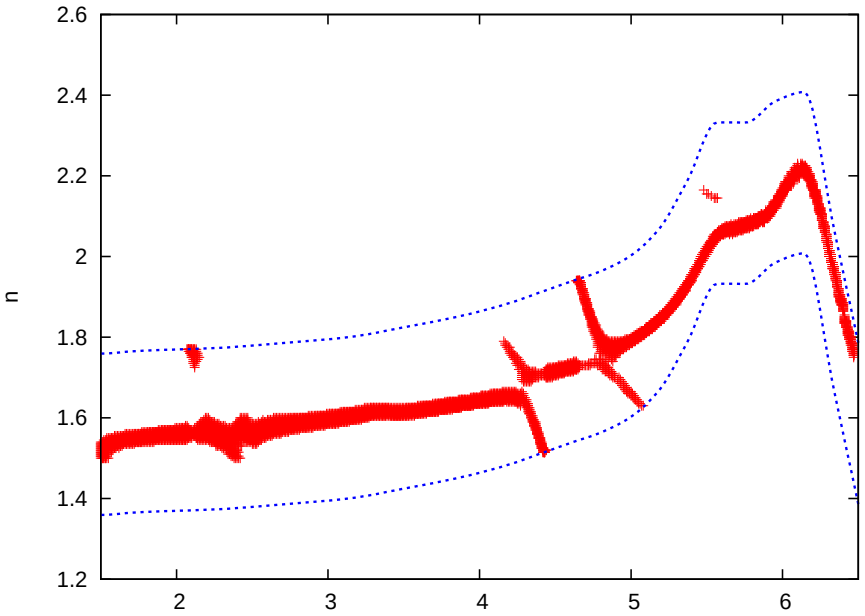
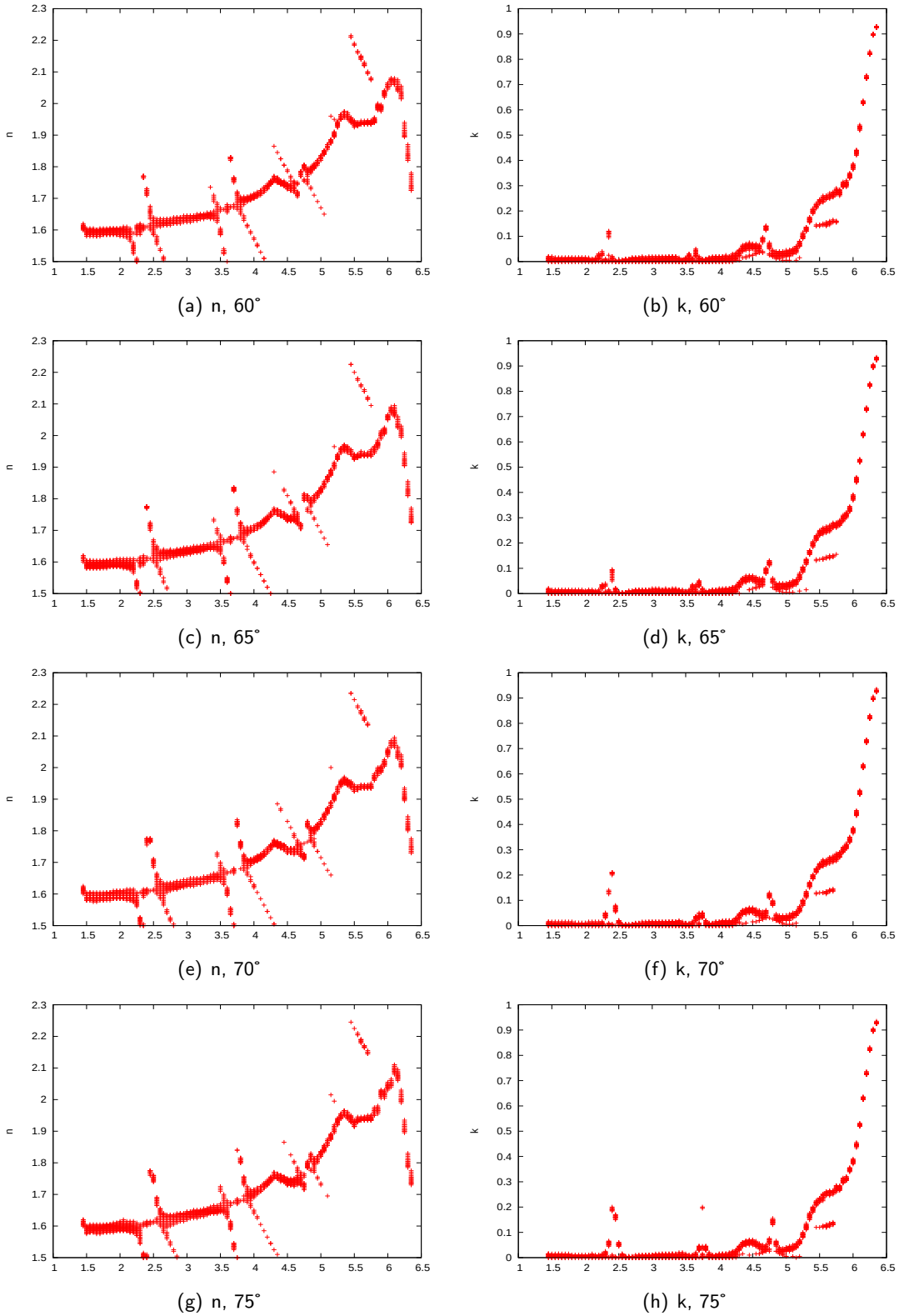


Fig. 11.3: Restriction de l'espace (n, k) pour trouver la solution physique.

On remarque qu'à l'issue de cette restriction, la continuité de la courbe de dispersion semble brisée à deux endroits : à 4,3eV et 4,8eV. Ceci est dû au fait que la solution, continue, passe localement par des endroits qui ne sont pas situés dans les minimums indiqués par les k-P.P.V.. La difficulté est donc de retrouver une courbe régulière malgré ces points délicats.

Deux moyens, éventuellement combinables, existent pour retrouver de la continuité :

- Ajuster l'angle d'incidence du rayon de lumière polarisée. En effet, il semble que cela "referme" la courbe aux points problématiques comme le montrent les graphiques 11.4 établis pour une résine dédiée à la lithographie par faisceau d'électron (*neb22*) ;
- Reconstruire les variations de (n, k) à la manière de ce qui a été présenté au chapitre 5 concernant la scatterométrie dynamique.



(a) n, 60° (b) k, 60°

(c) n, 65° (d) k, 65°

(e) n, 70° (f) k, 70°

(g) n, 75° (h) k, 75°

Fig. 11.4: Indices n et k d'une résine *neb22* trouvés par recherche des plus proches voisins, pour différentes valeurs de l'angle d'incidence θ. On remarque que l'angle 70° semble être le plus propice à la détermination des indices optiques puisque la continuité de la "trace" rouge y est la plus manifeste.

11.2 Reconstruction de la régularité

La reconstruction, on le rappelle, consiste à choisir, pour chaque énergie, le couple (n, k) qui, parmi les k éléments extraits de la bibliothèque, minimise la somme pondérée de :

- la proximité du couple de grandeurs ellipsométriques choisi dans la bibliothèque avec les mesures ;
- la continuité des valeurs de (n, k) tracées selon l'énergie.

$$\min_{(\mathbf{n}_i)_{1\leq i\leq N}} \left[\sum_{j=1}^{N} ||\tilde{\mathbf{D}}_j - \mathbf{D}_j||_1 + f((\mathbf{n}_i)_{1\leq i\leq N}) \right] \tag{11.1}$$

où $\tilde{\mathbf{D}}$ est la signature ellipsométrique mesurée de taille N. Il s'agit d'un problème de minimisation sur **la suite** $(\mathbf{n}_i)_{1\leq i\leq N}$ et non pas sur chacun des termes de cette suite pris indépendemment.

À la suite de cela, nous effectuons un lissage dans l'espace continu, indépendamment sur n et sur k.

Une différence notable existe entre l'application à la scatterométrie dynamique et aux indices optiques : si, dans le premier cas, la reconstruction est chronologique (on ne peut s'appuyer que sur les instants passés), il n'est pas justifié de procéder de la sorte ici. Le problème est de choisir les meilleurs valeurs de $\mathbf{n} = (n, k)$ pour chaque énergie en fonction des valeurs à toutes les autres énergies. Il s'agit de choisir la meilleure suite $(\mathbf{n}_i)_{1\leq i\leq N}$ parmi k^N possibilités.

Comme il n'est pas possible d'évaluer la fonction coût pour chacune des suites possibles à cause de leur nombre, on procédera "chronologiquement" tout de même, mais en faisant des aller-retours successifs (énergies croissantes, puis décroissantes, etc.). La fonctionnelle régularisante peut ainsi, dès le deuxième passage, s'appuyer sur les N choix de plus proches voisins déjà effectués pour affiner la solution globale.

Le choix de la fonctionnelle régularisante est toujours l'élément délicat car elle doit permettre la convergence de ce procédé ; c'est-à-dire qu'au bout d'un nombre fini d'aller-retours, les choix des N valeurs de $\mathbf{n}$ ne changent plus. Une recherche plus approfondie dans ce domaine devra être faite. Pour cet exemple de matériau polystyrène, nous avons utilisé comme fonctionnelle une combinaison de dérivées d'ordres différents (cf. tableau 11.1). Pour chaque terme, un vecteur-coefficient $\tilde{\beta}$ est à déterminer.

étape	**sens** (eV)	**fonctionnelle discrète**
1	↗	$\left\|\tilde{\beta_1} \cdot \frac{\mathbf{n}_i - \mathbf{n}_{i-1}}{\Delta_{h\nu}}\right\|^2$
2	↘	$\left\|\tilde{\beta_2} \cdot \frac{\mathbf{n}_i - \mathbf{n}_{i-20}}{\Delta_{h\nu}}\right\|^2 + \left\|\tilde{\beta_3} \cdot \frac{\mathbf{n}_{i-1} - 2\mathbf{n}_i + \mathbf{n}_{i+1}}{\Delta^2_{h\nu}}\right\|^2 + \left\|\tilde{\beta_4} \cdot \frac{\mathbf{n}_{i-2} - 3\mathbf{n}_{i-1} + 3\mathbf{n}_i - \mathbf{n}_{i+2}}{\Delta^3_{h\nu}}\right\|^2$
3	↗	idem étape 2
4-5-6	↘, ↗, ↘	idem étape 2

Tab. 11.1: Reconstruction de la linéarité des courbes de dispersion. À chaque étape, un nouveau $\mathbf{n}_i$ est choisi (parmi les k-P.P.V.) pour maximiser la régularité par rapport aux autres $\mathbf{n}_i$ choisis précédemment.

Le résultat final de ce procédé appliqué à la résine *poly28k* est tracé sur la figure 11.5. Pour la comparaison, on a tracé la courbe de dispersion du matériau à température ambiante (en pointillés rouges) : si la différence observée vers 4eV peut être due à un défaut de régularisation, on observe

cependant vers 5.5eV une différence qui n'est liée qu'à la température (ambiante pour la référence, et 140°pour notre exemple).

Le graphique du bas représente, comme au chapitre 9, le rapport de la régularisation : numéro du plus proche voisin et résidu associé. On observe ici que les croix rouges deviennent mieux réparties (sur les valeurs 1-16 PPV) à partir de 4eV environ, ce qui signifie que le procédé de régularisation a joué un rôle. En dessous de cette valeur, les courbes, déjà régulières, semblent ne pas nécessiter d'apport supplémentaire de régularité.

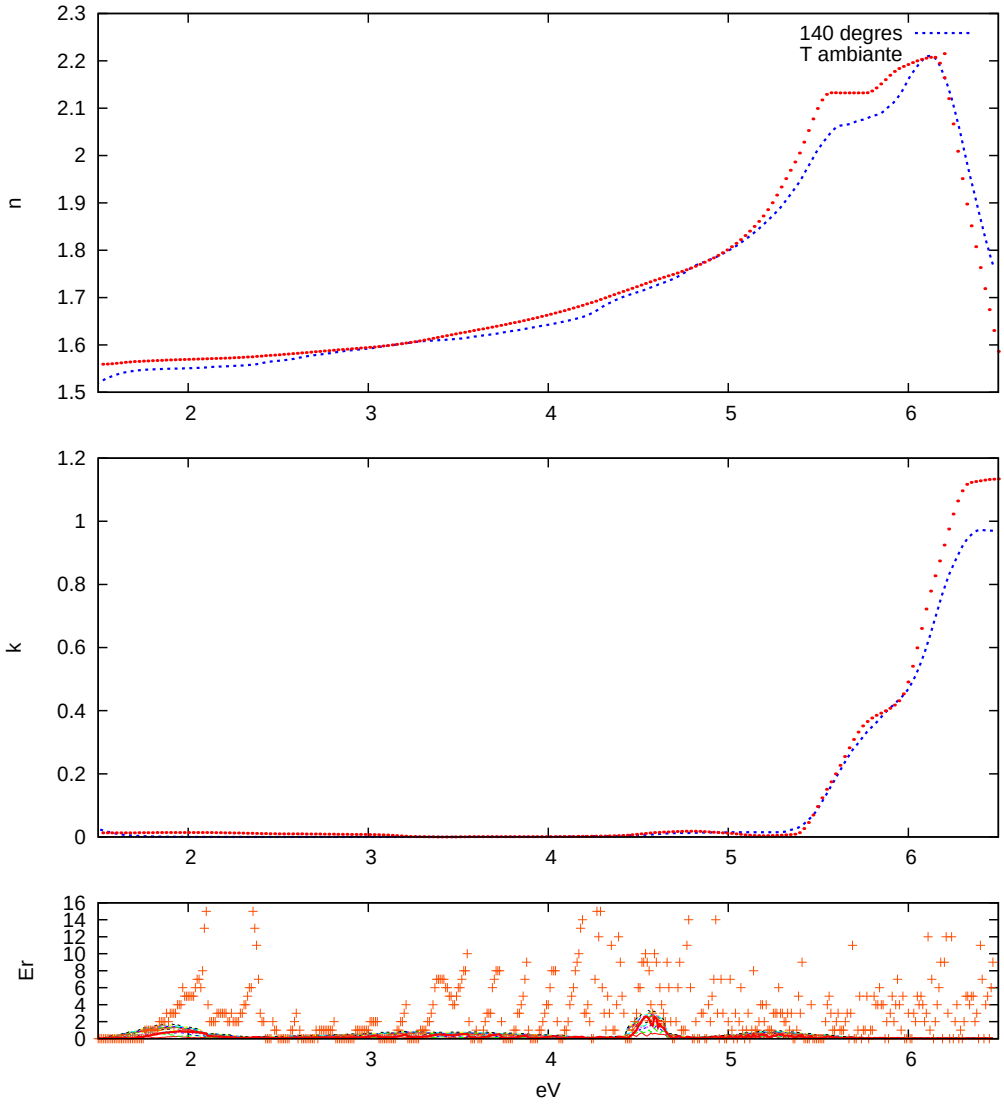


Fig. 11.5: Reconstruction de l'indice optique pour le matériau *poly28k*.

11.3 Conclusion : Utilisation comme conditions initiales

Par elle même, on l'a vu, la méthode de reconstruction développée pour la scatterométrie dynamique permet de déterminer efficacement les valeurs des indices optiques pourvu que de bonnes

fonctionnelles régularisantes soient trouvées, et cela, sans utiliser de loi de dispersion, ce qui est fondamental pour l'étude de matériaux au comportement optique complexe.

Mais elle peut aussi être utilisée pour fournir une estimation plus grossière des courbes de dispersion qui peuvent alors servir de conditions initiales à d'autres méthodes de détermination de nature itérative. Dans ce cas, le travail sur le mécanisme régularisant est moindre (mais pas nul) et c'est le coté visuel et intuitif de notre méthode (lors du gommage des parties de solution erronées) qui permet de prévenir les aberrations éventuelles qui surviennent lorsque ces autres méthodes convergent localement (en énergie) vers un minimum local (de l'espace (n, k)).

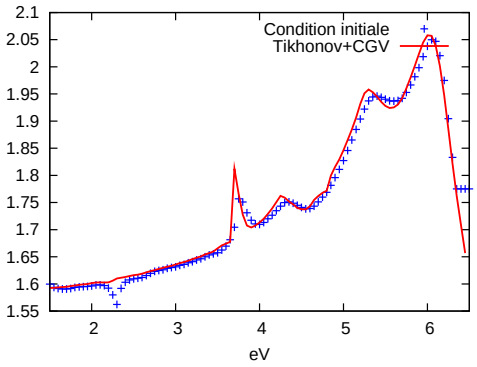
Cela s'illustre sur la figure 11.6 : nous avons tenté de déterminer les indices n et k du matériau *neb22* selon une méthode développée au laboratoire qui se base sur la régularisation de Tikhonov et sur la GCV (*Generalized Cross Validation* [22]). Cette méthode nécessite comme conditions initiales des estimations des courbes de dispersion afin de converger vers une solution valable plutôt que des minimums locaux.

Les graphiques 11.6(a) et 11.6(b) sont obtenus en utilisant notre méthode de la manière la plus grossière possible : nous n'avons pas cherché à régulariser les artefacts situés en 2,3eV et 3,8eV. Le premier de cet artefact est effacé lors de la méthode itérative alors que le second semble inscrire trop profondément la courbe dans un mauvais minimum local. Ceci justifie le soin à apporter tout de même à l'étape de régularisation de notre méthode.

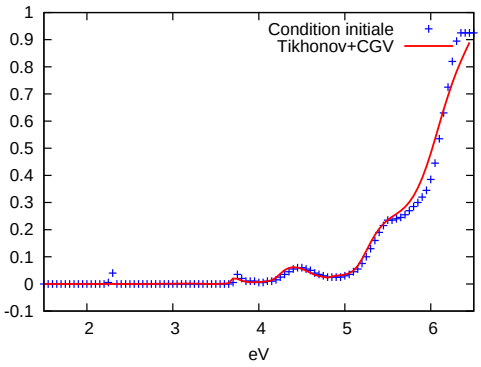
Les autres graphiques concernent l'utilisation comme condition initiale :

- d'un modèle plat $n = 1.6$, $k = 0$ pour 11.6(c) et 11.6(d).
- d'un modèle de Cauchy évalué sur la plage $1,5 - 4eV$ mais extrapolé sur $1,5 - 6,5eV$ pour 11.6(e) et 11.6(f).
- d'un modèle contenant un oscillateur pour 11.6(g) et 11.6(h).

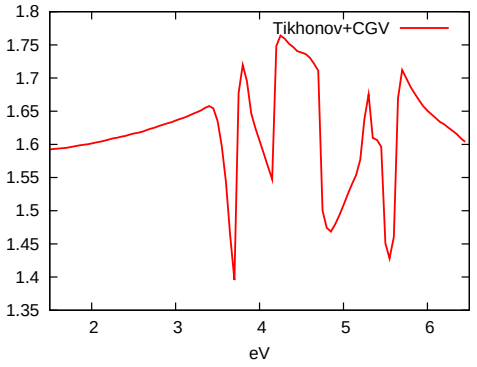
La combinaison de notre méthode et de la méthode itérative basée sur la GCV permet donc une détermination de courbes de dispersion pertinente : le contrôle des mauvais minimums locaux est effectué quasiment *de visu* et cela permet à l'algorithme itératif de démarrer avec de bonnes conditions.



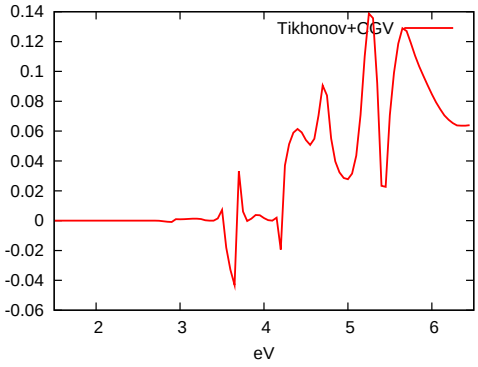
(a) n, conditions initiales : notre méthode



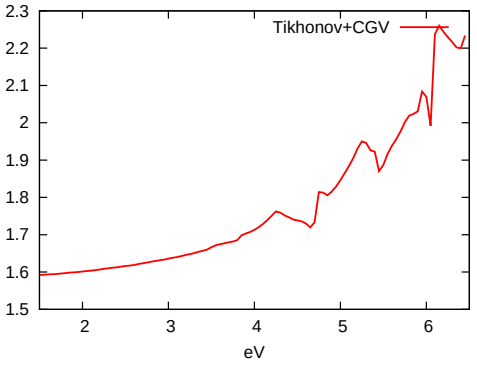
(b) k, conditions initiales : notre méthode



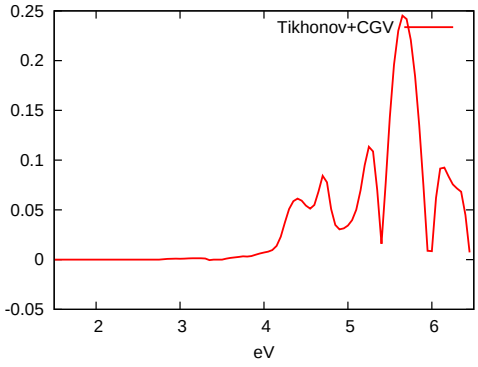
(c) n, conditions initiales : courbes plates



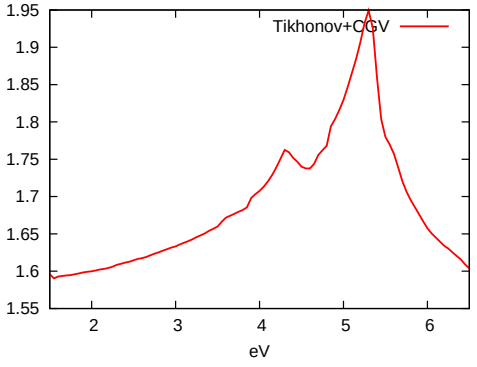
(d) k, conditions initiales : courbes plates



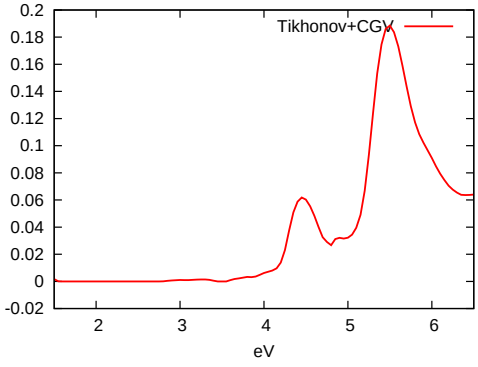
(e) n, conditions initiales : Cauchy



(f) k, conditions initiales : Cauchy



(g) n, conditions initiales : oscillateur



(h) k, conditions initiales : oscillateur

Fig. 11.6: Reconstruction de n et k par la méthode de Tikhonov, influence des conditions initiales.

Conclusion générale

Les progrès futurs de la microélectronique, notamment dans sa capacité à fabriquer des circuits toujours plus petits pour intégrer toujours plus de puissance, ne se feront pas sans une avancée considérable dans le champ de la métrologie, et notamment de la métrologie *in situ*, le but étant le suivi des procédés de fabrication en temps réel.

Pour cela, la scatterométrie possède des atouts considérables : c'est une technique non destructive et utilisable aisément *in situ*. Ces travaux de thèse ont permis d'initier le développement d'une version dynamique de la scatterométrie en proposant des solutions pour exploiter au mieux les données fournies par un ellipsomètre, ou plus généralement par tout système fournissant une signature optique caractéristique du motif diffractant (comme autre exemple : un réflectomètre). Les résultats obtenus sont encourageants et incitent à poursuivre les recherches.

Dans cette thèse, nous sommes partis d'un outil de métrologie développé pour des mesures statiques, la scatterométrie spectroscopique, pour l'appliquer au suivi en temps réel. Nous avons étudié les phénomènes physiques mis en jeux et la manière de les simuler numériquement, notamment par une méthode qui décompose le rayonnement électromagnétique en modes se propageant à travers un réseau diffractant.

Nous avons ensuite évalué les différentes manières de résoudre le problème inverse, c'est-à-dire d'utiliser les signatures scatterométriques acquises pour déterminer des grandeurs géométriques. Parmi ces méthodes nous avons choisi celle des bibliothèques, déjà éprouvée en milieu industriel et compatible avec la contrainte de temps réel. Cette contrainte signifie que la vitesse avec laquelle le problème inverse dynamique (reconstruction de la variation de la géométrie du profil diffractant) est résolu ne doit pas être inférieure à la vitesse avec laquelle les données sont rendues disponibles. En effet, si la reconstruction ne se fait pas dans les temps, il est impossible de procéder à un contrôle des conditions de réalisation de tel ou tel procédé.

Dans une troisième partie, nous avons développé la méthode des bibliothèques aux cas pour lesquels les signatures sont faiblement résolues mais où, en revanche, la fréquence d'acquisition est élevée (de l'ordre de 10 par seconde). Ces développements ont consisté, d'une part, en l'élaboration d'une méthode de régularisation (basée sur la méthode de Tikhonov) adaptée à ces bibliothèques et, d'autre part en l'utilisation inédite de processeurs dont l'architecture est dédiée au graphisme, les GPU.

Enfin, une fois ces développements réalisés, nous avons cherché à les valider par des expériences dont les ambitions à terme sont diverses : optimisation des procédés de nano-impression (fluage de résine) ou étude du comportement des motifs lors du *resist trimming* (gravure plasma).

Il s'est avéré en fin de thèse que la méthode de reconstruction pouvait s'appliquer de manière particulièrement efficace à la détermination des indices optiques d'un matériau. Cette détermination étant un point fondamental pour la précision de la scatterométrie, il apparaît que notre algorithme participe aussi au développement de cette technique de métrologie, *en général*, c'est-à-dire statique comme dynamique.

Ce travail de thèse constitue donc la première étape d'une évolution de la scatterométrie vers le temps réel. La poursuite de ces travaux s'effectuera dans le cadre d'une thèse orientée vers une application gravure plasma. La scatterométrie dynamique utilisée pour suivre une étape de gravure est en effet très attendue dans ce domaine car de nombreux phénomènes restent à ce jour mal compris. Le travail restant à réaliser est conséquent et les points à développer sont variés. Ils porteront notamment :

- D'un point de vue algorithmique, sur des développements de la méthode de reconstruction, et notamment sur les fonctionnelles régularisantes et leurs paramètres ;
- D'un point de vue instrumental, sur le développement d'un ellipsomètre (ou autre) ayant une cadence de mesure plus grande afin d'accéder à des procédés plus rapides. L'utilisation des GPU autorise d'ores et déjà un flux bien plus élevé mais les progrès qui pourront être réalisés sur l'outil de mesure restent un axe d'amélioration important ;
- D'un point de vue applications industrielles, sur l'adaptation de notre méthode à la réflectométrie, plus utilisée que la scatterométrie pour une utilisation directe sur les circuits ayant une nature périodique (mémoires, par exemple).

Ces points permettront d'étendre la validation et l'utilisation de la scatterométrie dynamique à des suivis de procédés que l'on sait complexes. Pour la gravure par plasma, des améliorations considérables pourront êtres apportées à la détermination des indices optiques (changeant en cours du procédé) ou à l'établissement du modèle géométrique.

Ces travaux pourront également porter sur d'autres procédés de fabrication de la microélectronique (dépôts de matériaux, autres lithographies, etc.) et même être étendus à d'autres domaines. Le travail réalisé au cours de cette thèse sur les processeurs graphiques est novateur et la puissance de cette architectures démontrée ici pour la recherche de k-P.P.V. ainsi que la méthode de reconstruction ouvrent des perspectives intéressantes dans différents domaines. Ils trouveraient probablement une application en imagerie médicale (le problème inverse serait par exemple issu de la tomographie), en vision assistée par ordinateur (détection et suivi de formes en temps réel), etc.

ANNEXE A

Recommandations de l'ITRS

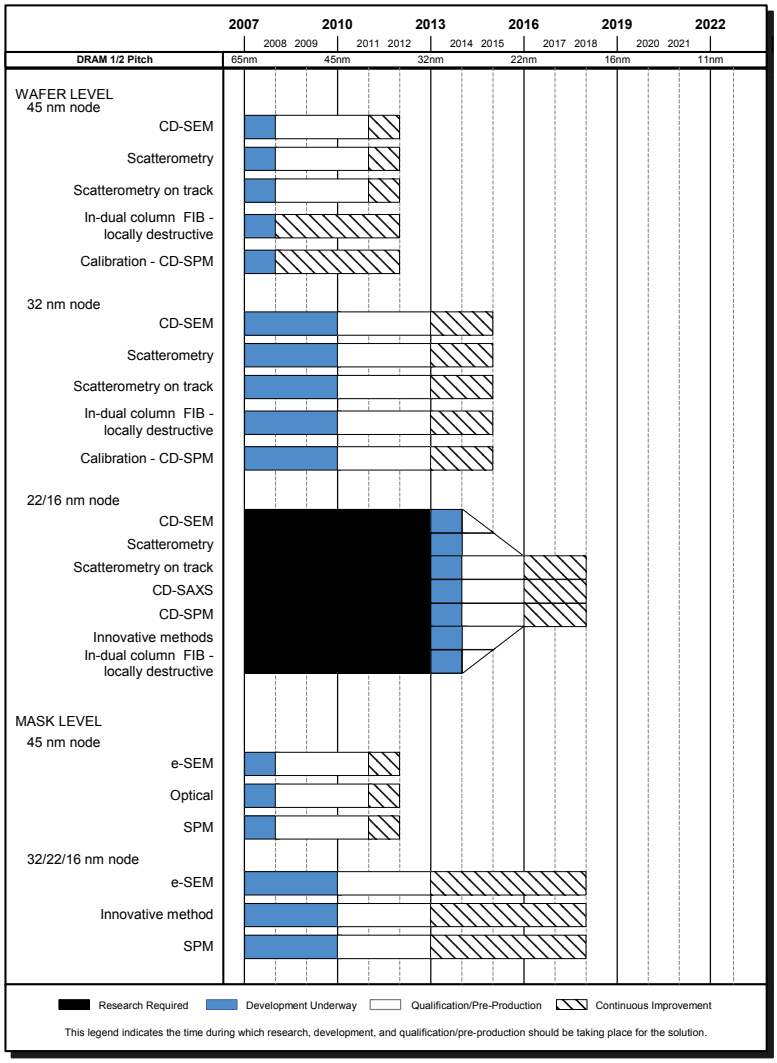


Figure MET2a Lithography Metrology Potential Solutions: CD

THE INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS: *2007*

Fig. A.1: Estimation de l'ITRS des technologies de métrologie pertinentes pour la mesure de la dimension critique dans les années à venir.

ANNEXE B

Méthode modale par développement de Fourier

La méthode présentée ici vise à simuler rigoureusement la réponse d'un motif périodique à une excitation magnétique. Elle fut mise au point dans les années 80 par Gaylord, Moharam *et al.*[29] puis régulièrement améliorée depuis. Son principe de base, détaillé dans cette annexe, consiste essentiellement en trois points :

- Le motif périodique est décomposé en N couches telles que la permittivité diélectrique soit uniforme selon z et périodique selon x. Ceci permet de décomposer cette permittivité en série de Fourier et les champs sur une base de Floquet.
- L'introduction des composantes de ces champs dans les équations de propagation mène à une suite de problèmes aux valeurs propres dont les solutions sont les modes de propagation.
- Les efficacités de diffraction (grandeurs recherchées) sont obtenues après avoir relié ces modes par les relations de passage (continuités des composantes tangentielles des champs électrique et magnétique) .

La description de la méthode MMFE se fera dans un premier temps par le cas élémentaire du réseau lamellaire (une seule couche). Ensuite, sera détaillée la généralisation à N couches.

B.1 Problème élémentaire

Nous modéliserons ici le phénomène de diffraction d'une onde lumineuse par un réseau lamellaire (mono-couche, tel que représenté sur le schéma B.1) de manière à obtenir les efficacités de diffraction de chacun des ordres réfléchis et transmis. Ce problème élémentaire sera étendu ensuite aux cas des réseaux dont les géométries sont plus complexes et qui seront modélisés par un empilement de plusieurs couches.

Le système diffractant est donc constitué de trois domaines :

- **domaine I** ($z < 0$) : air. La permittivité diélectrique relative complexe est uniforme et semblable au vide $\epsilon_I = 1$
- **domaine II** ($0 < z < d$) : motif diffractant de période Λ_x selon x, invariant selon y et z. La permittivité diélectrique (comme son inverse) peut donc se décomposer en série de Fourier.
- **domaine III** ($z > d$) : substrat de permittivité complexe connue ϵ_{III}

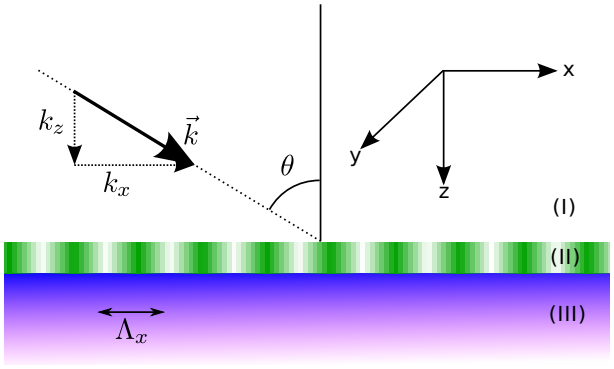


Fig. B.1: Système diffractant périodique monocouche

L'onde incidente est une onde plane monochromatique de longueur d'onde dans le vide λ ; elle se propage dans le plan (xOz) et arrive du domaine I vers le domaine II avec un angle d'incidence θ.

La propagation se faisant dans le plan (xOz), ce problème présente une invariance selon l'axe y et cela engendre le découplage des modes TE et TM dans les régions uniformes comme dans la région modulée[1]. Dans le cas contraire, on parle d'*incidence conique* (cf. schéma B.2) et la résolution du problème se trouve complexifiée [60] [32].

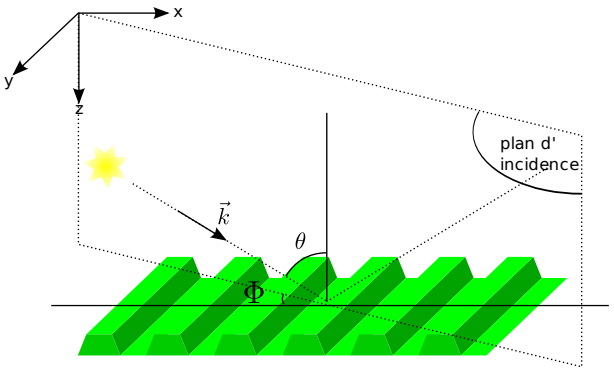


Fig. B.2: Incidence conique d'un rayon lumineux sur un réseau de lignes : le plan d'incidence n'est pas perpendiculaire aux lignes du réseau.

D'autre part, le régime considéré étant harmonique de pulsation $\omega = 2\pi c/\lambda$, on retirera dans la suite, pour plus de clarté, le facteur $e^{i\omega t}$ des expressions.

Tout cela considéré, on peut exprimer l'onde incidente, d'amplitude unitaire, de cette manière :

$$U_0(x,z) = e^{-i\tilde{n}_I(k_x x + k_z z)} = e^{-i\tilde{n}_I k_0(x\sin\theta + z\cos\theta)}$$

où $k_0 = \frac{2\pi}{\lambda}$ est le nombre d'onde de l'onde incidente dans le vide, et :

$$U_0 = \begin{cases} E_y & \text{dans le cas d'une polarisation TE} \\ H_y & \text{dans le cas d'une polarisation TM} \end{cases}$$

[1] Pour s'en convaincre, il suffit de projeter les équations de Maxwell 1.17 et 1.19 sur les axes du repère (x, y, z) et d'annuler ∂_y : on s'aperçoit alors que les composantes TE (E_y , H_x et H_z) sont indépendantes des composantes TM (H_y, E_x et E_z).

B.1.1 Champs dans les régions homogènes

Les régions I et III sont des régions où la permittivité diélectrique relative complexe est constante sur tout l'espace. La description des champs harmoniques est donc réalisée par l'équation d'Helmholtz (cf. éq. 1.20) :

$$\left(\nabla^2 + k_0^2 \tilde{\epsilon}_r\right) \mathbf{U} = 0 \tag{B.1}$$

D'autre part, nous avons vu que la présence d'un réseau faisait diffracter la lumière et que cette diffraction était une répartition de la puissance lumineuse en un certain nombre d'ordres réfléchis et transmis (cf. section 1.3) qui ont tous une direction de propagation propre. Nous avions alors noté que les composantes selon x des vecteurs d'ondes des modes m réfléchis et transmis étaient identiques (éq. 1.22) :

$$\alpha_m = \tilde{n} k_0 \sin\theta + mK_x$$

Nous pouvons ainsi déduire les composantes selon z des modes m réfléchis et transmis, r_m et t_m (Théorème de Pythagore, le vecteur d'onde $\mathbf{k}$ étant l'hypoténuse) :

$$\begin{cases} \alpha_m^2 + r_m^2 = k_I^2 \\ \alpha_m^2 + t_m^2 = k_{III}^2 \end{cases}$$

Le champ total dans les régions homogènes est donc :

- Pour la région I : onde incidente U_0 plus le champ réfléchi décomposé en modes de diffraction (U_R)
- Pour la région III : champ transmis U_T, lui aussi décomposé en modes

Ce qui, formellement, s'écrit :

$$\begin{cases} U_I(x,z) = U_0(x,z) + U_R(x,z) \\ \qquad\qquad = e^{-i[\alpha_0 x + r_0 z]} + \sum_m R_m e^{-i[\alpha_m x - r_m z]} & \text{(B.2)} \\ U_{III}(x,z) = U_T(x,z) \\ \qquad\qquad = \sum_m T_m e^{-i[\alpha_m x + t_m (z-d)]} & \text{(B.3)} \end{cases}$$

où :

- U_I, U_{III} sont les champs totaux des régions I et III
- R_m et T_m les amplitudes complexes de l'ordre de diffraction m respectivement réfléchi et transmis.

Cette expression des champs est aussi appelée **développement de Rayleigh** [83] (ou développement en ondes planes) ; la base de Rayleigh est l'ensemble des fonctions $e^{-i\mathbf{k}_m \cdot \mathbf{r}}$ correspondant ici aux différents modes diffractés.

La résolution de notre problème consistera à déterminer les **efficacités de diffraction** des modes réfléchis et transmis. Il s'agit du rapport de l'énergie propagée par chacun des modes sur l'énergie incidente[2]. Ces grandeurs sont données par :

[2] L'énergie est ici calculée comme la projection selon z (le système étant périodique en x) de la moyenne temporelle du vecteur de Poynting $\Pi = \mathbf{E} \times \mathbf{H}$.

$$\begin{cases} \eta_m^R = Re\left(\frac{r_m}{r_0}\right)|R_m|^2 & \text{(B.4)} \\ \eta_m^T = Re\left(C\frac{t_m}{t_0}\right)|T_m|^2 & \text{(B.5)} \end{cases}$$

où :

$$C = \begin{cases} 1 & \text{pour le mode TE} \\ \left(\frac{n_I}{n_{III}}\right)^2 & \text{pour le mode TM} \end{cases} \quad \text{(B.6)}$$

Afin de rendre possible la résolution numérique du problème, nous tronquerons les valeurs de m à $[-M, \cdots, +M]$. Nous aurons ainsi $2 \times (2M+1)$ inconnues : R_m et T_m.

B.1.2 Champs dans la région modulée

La région II **n'est pas homogène**. Cela signifie que la permittivité diélectrique relative n'est plus uniforme, mais dépend de la coordonnée d'espace x : $\epsilon_{II}(x)$. L'équation de propagation en milieu homogène B.1 n'est donc plus valable et il convient maintenant d'en dériver de nouvelles en repartant des équations de Maxwell (cf. section 1.1.1).

Les modes TE et TM étant découplés dans notre cas, nous pouvons écrire des équations indépendantes pour les champs transverses $E_y(x,z)$ et $H_y(x,z)$:

Pour le mode transverse électrique : $U(x,z) = E_y(x,z)$

$$\left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} + k^2\epsilon(x,z)\right] E_y(x,z) = 0 \quad \text{(B.7)}$$

Pour le mode transverse magnétique : $U(x,z) = H_y(x,z)$

$$\frac{\partial}{\partial x}\left[\frac{1}{\epsilon(x,z)}\frac{\partial H_y(x,z)}{\partial x}\right] + \frac{\partial}{\partial z}\left[\frac{1}{\epsilon(x,z)}\frac{\partial H_y(x,z)}{\partial z}\right] + k_0^2 H_y(x,z) = 0 \quad \text{(B.8)}$$

B.1.3 Conditions aux limites

Les relations de passage 1.10 et 1.11 indiquent que les composantes tangentielles des champs électrique et magnétique sont continues au passage d'une interface située en $z = z_i$.

Ainsi, pour la polarisation TE, on a[3] :

$$E_y(x, z_i^-) = E_y(x, z_i^+) \quad \text{(B.9)}$$

$$H_z(x, z_i^-) = H_z(x, z_i^+) \Rightarrow \frac{\partial E_y}{\partial z}(x, z_i^-) = \frac{\partial E_y}{\partial z}(x, z_i^+) \quad \text{(B.10)}$$

Et pour la polarisation TM :

$$H_y(x, z_i^-) = H_y(x, z_i^+) \quad \text{(B.11)}$$

$$E_z(x, z_i^-) = E_z(x, z_i^+) \Rightarrow \frac{1}{\epsilon(x, z_i^-)}\frac{\partial H_y}{\partial z}(x, z_i^-) = \frac{1}{\epsilon(x, z_i^+)}\frac{\partial H_y}{\partial z}(x, z_i^+) \quad \text{(B.12)}$$

À ce niveau, il reste à mettre en relation les deux équations d'Helmholtz B.7 et B.8 avec les conditions aux limites B.9, B.10, B.11 et B.12.

[3] On montre les implications B.10 et B.12 en s'aidant des équations de Maxwell.

B.1.4 Modes propres pour les champs TE

La région II est caractérisée par une permittivité diélectrique périodique, donc décomposable en série de Fourier :

$$\epsilon_{II}(x) = \sum_{p=-\infty}^{p=+\infty} \epsilon_p(z) e^{-ipK_x} \tag{B.13}$$

où $K_x = \frac{2\pi}{\Lambda_x}$

De plus, à l'intérieur de cette région modulée, on peut développer le champ électrique sur une base de Floquet-Bloch [4] :

$$E_y(x,z) = \sum_m S_m(z) e^{-i\alpha_m x} \tag{B.14}$$

En incluant cette expression dans l'équation de propagation B.7, il vient :

$$-\sum_m S_m(z)\alpha_m^2 e^{-i\alpha_m x} + \sum_m \frac{\partial^2 S_m(z)}{\partial z^2} e^{-i\alpha_m x} + k_0^2 \left(\sum_p \epsilon_p e^{ipK_x x}\right)\left(\sum_m S_m(z) e^{-i\alpha_m x}\right) = 0$$

On multiplie la précédente expression par $e^{i\alpha_l x}$, $(\forall l \in \mathbb{Z})$ et on l'intègre ensuite sur une période Λ_x [5]. On obtient alors :

$$\frac{\partial^2 S_l(z)}{\partial z^2} = \sum_m (\alpha_m^2 \delta_{lm} - k_0^2 \epsilon_{l-m}) S_m(z) \tag{B.15}$$

où δ_{lm} est le symbole de Kronecker (valant 1 si $l = m$, 0 sinon). C'est un système linéaire qui se symbolise par :

$$\ddot{\mathcal{S}} = \mathcal{M}\mathcal{S} \tag{B.16}$$

où la matrice $\mathcal{M}$ définie ainsi

$$\mathcal{M}_{lm} = \alpha_m^2 \delta_{lm} - k_0^2 \epsilon_{l-m}$$

est la somme d'une matrice diagonale contenant les α_m et d'une matrice de Toeplitz (matrice à diagonales constantes) contenant les composantes de la série de Fourier B.13

On tronque le système à $2M+1$ éléments, $[-M, \cdots, +M]$, et on va chercher à diagonaliser $\mathcal{M}$, c'est-à-dire à trouver une matrice de vecteurs propres $\mathcal{V}$ et une suite de valeurs propres $(r_m^2)_{-M \leq m \leq +M}$ rangée dans une matrice diagonale $[[r^2]]$ telles que :

$$\mathcal{M} = \mathcal{V}[[r^2]]\mathcal{V}^{-1} \tag{B.17}$$

On pose ensuite $\mathcal{T} = \mathcal{V}^{-1}\mathcal{S}$. En multipliant à gauche le système B.16 par $\mathcal{V}^{-1}$ il vient ce système simple d'équations différentielles du second degré :

[4] Le théorème sous-jacent a été énoncé en premier en 1883 par Gaston Floquet [24] puis énoncé à nouveau pour la cristallographie en 1928 par le suisse Félix Bloch [17]. Une démonstration pour le cas des équations de Maxwell peut être trouvée en référence [89].

[5] On utilise ici le fait que $\int_{\Lambda_x} e^{i\alpha_l x} e^{-i\alpha_m x} = 0$ si $l \neq m$.

$$\ddot{\mathcal{T}} = \begin{bmatrix} r_{-M}^2 & & \\ & \ddots & \\ & & r_{+M}^2 \end{bmatrix} \mathcal{T}$$

Les solutions d'un tel système sont :

$$T_m(z) = C_{1m} e^{-r_m z} + C_{2m} e^{r_m z} \tag{B.18}$$

où les C_{1m} et C_{2m} sont des nouvelles valeurs à déterminer (en plus des R_m et T_m).

Finalement, la solution du système B.16 est $\forall l \in [-M, \cdots, +M]$:

$$S_l(z) = \sum_{m=-M}^{+M} V_{lm}(C_{1m} e^{-r_m z} + C_{2m} e^{r_m z}) \tag{B.19}$$

où V_{lm} est l'élément d'indices (l, m) de la matrice de vecteurs propres $\mathcal{V}$. Les valeurs propres issues de la diagonalisation de $\mathcal{M}$ sont des carrés (r_m^2). Donc r_m tout comme son opposé sont solutions. On choisira pour la suite les valeurs r_m de cette manière :

$$r_m = \begin{cases} \sqrt{r_m^2} & \text{si } r_m^2 \text{ est réel} \\ \sqrt{r_m^2} \text{ avec } I(r_m) > 0 & \text{si } r_m^2 \text{ est complexe} \end{cases} \tag{B.20}$$

Cela permet de considérer l'équation B.19 comme la somme d'un terme correspondant à un mode se propageant vers les z croissants $(e^{-r_m z})$ et d'un terme correspondant à un mode se propageant vers les z décroissants. Le choix d'une partie imaginaire positive ou nulle est justifié par le fait qu'aucun des modes ascendants (c'est-à-dire, avec notre convention, se dirigeant vers les z décroissants) ne doit avoir une intensité exponentiellement croissante.

Au final, il s'agit de résoudre les équations suivantes (équations B.2, B.3, B.14 et B.19) :

$$E_{Iy}(x, z) = E_{0y}(x, z) + \sum_m R_m e^{-i[\alpha_m x - r_m z]} \tag{(B.2)}$$

$$E_{III y}(x, z) = \sum_m T_m e^{-i[\alpha_m x + t_m (z-d)]} \tag{(B.3)}$$

$$E_y(x, z) = \sum_l S_l(z) e^{-i\alpha_l x} \tag{(B.14)}$$

$$S_l(z) = \sum_m V_{lm}(C_{1m} e^{-r_m z} + C_{2m} e^{r_m z}) \tag{(B.19)}$$

et pour cela il nous reste à déterminer $4 \times (2M+1)$ variables $(R_m,\ T_m,\ C_{1m},\ C_{2m})_{-M \leq m \leq +M}$ en utilisant les $4 \times (2M+1)$ conditions aux limites données par les relations de passages en $z = 0$ et $z = d$ (continuités des composantes tangentielles des deux champs).

Continuité de $E_y(x, z)$ et $\frac{\partial E_y}{\partial z}(x, z)$ en $z = 0$

On introduit les équations B.2, B.14 et B.19 dans B.9 et B.10 :

$$\begin{cases} E_{Iy}(x, 0) = E_{IIy}(x, 0) \\ \dfrac{\partial E_{Iy}}{\partial z}(x, 0) = \dfrac{\partial E_{IIy}}{\partial z}(x, 0) \end{cases}$$

$$\begin{cases} e^{-i\alpha_0 x} + \sum_l R_l e^{-i\alpha_l x} = \sum_l S_l(z) e^{-i\alpha_l x} \\ \qquad\qquad\qquad\qquad = \sum_l \sum_m V_{lm}(C_{1m} + C_{2m}) e^{-i\alpha_l x} \\ -ir_0 e^{-i\alpha_0 x} + \sum_l ir_l R_l e^{-i\alpha_l x} = \sum_l \dfrac{\partial S_l(z)}{\partial z} e^{-i\alpha_l x} \\ \qquad\qquad\qquad\qquad = \sum_l \sum_m V_{lm}(-r_m C_{1m} + r_m C_{2m}) e^{-i\alpha_l x} \end{cases}$$

À nouveau, en multipliant par $e^{i\alpha_l x}$ puis en intègrant sur Λ_x :

$$\begin{cases} \delta_{0l} + R_l = \sum_m V_{lm}(C_{1m} + C_{2m}) \\ -ir_l(\delta_{0l} - R_l) = \sum_m V_{lm} r_m(-C_{1m} + C_{2m}) \end{cases}$$

Ce qui peut s'écrire sous la forme matricielle suivante :

$$\mathcal{W}_0 \begin{bmatrix} \delta_0 \\ C_2 \end{bmatrix} = \mathcal{W}_1 \begin{bmatrix} R \\ C_1 \end{bmatrix} \tag{B.21}$$

avec :

$$\begin{cases} \mathcal{W}_0 = \begin{bmatrix} I & -\mathcal{V} \\ -ir & \mathcal{V}r \end{bmatrix} & \text{(B.22)} \\ \mathcal{W}_1 = \begin{bmatrix} -I & \mathcal{V} \\ -ir & +\mathcal{V}r \end{bmatrix} & \text{(B.23)} \end{cases}$$

où :

- I est la matrice identité
- $\mathcal{V}$ est la matrice des éléments V_{lm}
- r la matrice diagonale d'éléments r_m (idem pour les matrices R, T, C_1, C_2 et δ_0.)

On peut ainsi exprimer le système à l'aide d'une **matrice d'interface $\mathbf{s}^0$** :

$$\begin{bmatrix} R \\ C_1 \end{bmatrix} = \mathbf{s}^0 \begin{bmatrix} \delta_0 \\ C_2 \end{bmatrix} \tag{B.24}$$

où $\mathbf{s}^0 = \mathcal{W}_1^{-1}\mathcal{W}_0$, il s'agit de la matrice qui lie le rayonnement sortant de l'interface en $z = 0$, R et C_1, aux modes entrants : δ_0 et C_2 (cf. fig. B.3).

Cette matrice (comme toutes les autres de ce type que nous rencontrerons dans la suite) se décompose en quarts liant les rayonnements progressant vers le bas (dans le sens z positif, d : *down*) à ceux progressant vers le haut (sens s négatif, u : *up*) :

$$\mathbf{s}^0 = \begin{bmatrix} (\mathbf{s}_0)_{du} & (\mathbf{s}_0)_{uu} \\ (\mathbf{s}_0)_{dd} & (\mathbf{s}_0)_{ud} \end{bmatrix}$$

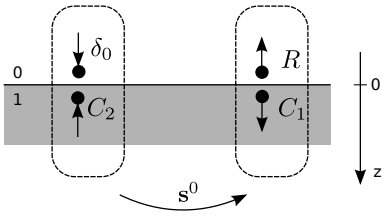


Fig. B.3: Première matrice d'interface

Continuité de $E_y(x,z)$ et $\frac{\partial E_y}{\partial z}(x,z)$ en $z=d$

De la même manière, on introduit les équations B.3, B.14 et B.19 dans B.9 et B.10 :

$$\begin{cases} E_{Iy}(x,d) = E_{IIy}(x,d) \\ \dfrac{\partial E_{Iy}}{\partial z}(x,d) = \dfrac{\partial E_{IIy}}{\partial z}(x,d) \end{cases}$$

$$\begin{cases} \displaystyle\sum_l T_l e^{-i\alpha_l x} = \sum_l S_l(z) e^{-i\alpha_l x} \\ \qquad\qquad = \displaystyle\sum_l \sum_m V_{lm}(C_{1m}e^{-r_m d} + C_{2m}e^{r_m d})e^{-i\alpha_l x} \\ \displaystyle\sum_l i t_l T_l e^{-i\alpha_l x} = \sum_l \left.\frac{\partial S_l(z)}{\partial z}\right|_{z=d} e^{-i\alpha_l x} \\ \qquad\qquad = \displaystyle\sum_l \sum_m V_{lm} r_m(-C_{1m}e^{-r_m d} + C_{2m}e^{r_m d})e^{-i\alpha_l x} \end{cases} \tag{B.25}$$

À nouveau, en multipliant par $e^{i\alpha_l x}$ puis en intégrant sur Λ_x il vient :

$$\begin{cases} T_l = \displaystyle\sum_m V_{lm}(C_{1m}e^{-r_m d} + C_{2m}e^{r_m d}) \\ -i t_l T_l = \displaystyle\sum_m V_{lm} r_m(-C_{1m}e^{-r_m d} + C_{2m}e^{r_m d}) \end{cases}$$

Ce qui peut s'écrire sous la forme matricielle suivante ($\mathcal{W}_1$ étant la matrice définie, éq. B.23) :

$$\mathcal{W}_1 \Omega^- \begin{bmatrix} C_1 \\ 0 \end{bmatrix} = \mathcal{W}_2 \Omega^+ \begin{bmatrix} C_2 \\ T \end{bmatrix} \tag{B.26}$$

avec :

$$\begin{cases} \Omega^+ = \begin{bmatrix} diag(e^{r_m d}) & 0 \\ 0 & I \end{bmatrix} \\ \Omega^- = \begin{bmatrix} I & 0 \\ 0 & diag(e^{-r_m d}) \end{bmatrix} \\ \mathcal{W}_2 = \begin{bmatrix} \mathcal{V} & -I \\ \mathcal{V}r & -it \end{bmatrix} \end{cases}$$

où t est la matrice diagonale d'éléments t_m.

Ce système s'exprime aussi sous la forme d'une **matrice de couche** $\tilde{s}^1$:

$$\begin{bmatrix} C_2 \\ T \end{bmatrix} = \tilde{s}_1 \begin{bmatrix} C_1 \\ 0 \end{bmatrix} \tag{B.27}$$

où :

$$\begin{aligned} \tilde{\mathbf{s}}^1 &= (\Omega^+)^{-1}\mathcal{W}_2^{-1}\mathcal{W}_1 \\ &= (\Omega^+)^{-1}\mathbf{s}^0\Omega^- \end{aligned} \tag{B.28}$$

est la matrice liant les modes sortants de la première couche (C_2, T) aux modes entrants (C_1 seulement car il n'y a pas de rayonnement venant du bas) ; ceci est représenté sur la figure B.4.

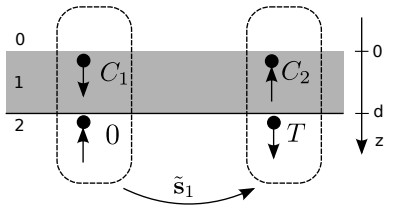


Fig. B.4: Première matrice de couche

'A ce point, il reste à combiner les relations et pour obtenir la relation finale entre l'onde incidente et les modes diffractés et transmis par ce réseau d'une couche :

$$\begin{bmatrix} R \\ T \end{bmatrix} = \mathbf{S}^1 \begin{bmatrix} \delta_0 \\ 0 \end{bmatrix} \tag{B.29}$$

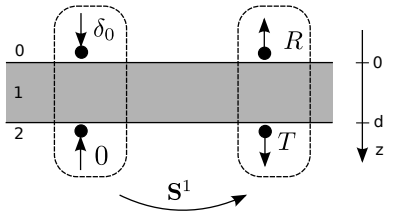


Fig. B.5: Première matrice S, $\mathbf{S}^1$

$\mathbf{S}^1$ s'obtient à l'aide de la matrice d'interface $\mathbf{s}^0$ et de la matrice de couche $\tilde{\mathbf{s}}_1$. Sa partie gauche (la droite est facteur de 0, donc quelconque) s'exprime ainsi :

$$\begin{cases} R = (\mathbf{S}^1)_{du} = [(\mathbf{s}^0)_{du} + (\mathbf{s}^0)_{uu}(\tilde{\mathbf{s}}^1)_{du}[1 - (\mathbf{s}^0)_{ud}(\tilde{\mathbf{s}}^1)_{du}]^{-1}(\mathbf{s}^0)_{dd}]\delta_0 \\ T = (\mathbf{S}^1)_{dd} = [(\tilde{\mathbf{s}}^1)_{dd}[1 - (\mathbf{s}^0)_{ud}(\tilde{\mathbf{s}}^1)_{du}]^{-1}(\mathbf{s}^0)_{dd}]\delta_0 \end{cases}$$

B.1.5 Modes propres pour les champs TM

Pour le mode transverse magnétique, nous procéderons de manière analogue au mode transverse électrique : il s'agira de résoudre l'équation de propagation B.8 en utilisant les développements de Fourier et de Floquet-Bloch suivants [6] :

[6] Dans cette section consacrée au mode TM, nous noterons les grandeurs homologues à celles de la section TE de manière identique. Ceci permet de ne pas alourdir la notation et de signifier que, dans le code électromagnétique, ce

$$\begin{cases} \dfrac{1}{\epsilon(x)} = \sum_p \tilde{\epsilon}_p e^{-ipK_x x} & \text{(B.30)} \\ H_y(x) = \sum_m S_m(z) e^{-i\alpha_m x} & \text{(B.31)} \end{cases}$$

Après les mêmes manipulations que précédemment (multiplication par $e^{i\alpha_I x}$ ($I \in \mathbb{Z}$) et intégration sur une période Λ_x), il vient :

$$\alpha_I \sum_p (\tilde{\epsilon}_{I-p} \alpha_p S_p(z)) - \sum_p \left[\tilde{\epsilon}_{I-p} \frac{\partial^2 S_p(z)}{\partial z^2} \right] - k_0^2 S_I(z) = 0 \quad \text{(B.32)}$$

ce qui correspond, matriciellement, à :

$$\ddot{\mathcal{S}} = \mathcal{M}\mathcal{S} \quad \text{(B.33)}$$

où, dans ce cas TM, $\mathcal{M}$ s'exprime à l'aide des matrices diagonales contenant les $\tilde{\epsilon}_m$ et α_m ($m \in [-M, \cdots, +M]$) notées $[[\tilde{\epsilon}]]$ et $[[\alpha]]$:

$$\mathcal{M} = [[\tilde{\epsilon}]]^{-1}([[\alpha]][[\tilde{\epsilon}]][[\alpha]] - k_0^2 \mathcal{I})$$

Dans le but d'améliorer significativement la convergence du calcul MMFE, une nouvelle formulation de l'expression précédente a été proposée (cf : Lalanne *et al.* [48] G. Granet *et al.*[33] et Lifeng Li *et al.*[59]) dans laquelle $[[\tilde{\epsilon}]]$ est remplacé par $[[\epsilon]]^{-1}$.

La solution du système B.33 est :

$$S_I(z) = \sum_m V_{Im}(C_{1m} e^{-\lambda_m z} + C_{2m} e^{\lambda_m z}) \quad \text{(B.34)}$$

Comme précédemment, le problème consiste maintenant à déterminer les $2 \times (2M+1)$ constantes C_{1m} et C_{2m} en utilisant les conditions de continuité aux interfaces. On arrive exactement aux mêmes systèmes de matrices que pour le mode TE. La généralisation de ce mode de calcul à plusieurs couches que nous allons expliquer maintenant est donc valable tout autant pour le mode TE que TM.

B.2 Généralisation à une structure multicouche

Le calcul d'efficacité de diffraction en utilisant la MMFE suppose une modélisation du motif sous forme de couches d'épaisseurs variables dans lesquelles la permittivité ϵ ne dépend plus que de x (cf. schéma 3.4). Il s'agit donc, pour le mode TE comme pour le mode TM, de combiner une succession de matrices de couches $\tilde{\mathbf{s}}$ pour obtenir la matrice finale $\mathbf{S}^N$ (si le réseau a N couches).

Dans le but de généraliser ce qui a été expliqué précédemment, on définit les vecteurs $\mathbf{u}^p(z)$ et $\mathbf{d}^p(z)$ comme les amplitudes complexes des modes ascendants et descendants à l'intérieur de la couche p du réseau. Ces modes sont représentés sur la figure B.6.

sont les mêmes instructions qui sont utilisées. Mais en aucun cas, ces grandeurs sont égales. Les grandeurs en questions sont : S_m, V_{lm}, C_{1m}, C_{2m} et r_m.

$$\begin{cases} d_m^p(z) = C_{1m}^p e^{-r_m^p(z-z^{p-1})} & \text{(B.35)} \\ u_m^p(z) = C_{2m}^p e^{r_m^p(z-z^{p-1})} & \text{(B.36)} \end{cases}$$

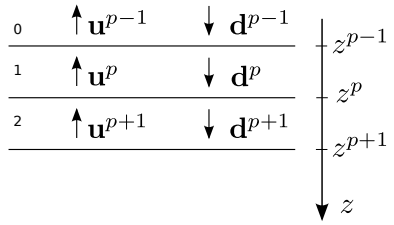


Fig. B.6: Modes ascendants ($\mathbf{u}^p$) et descendant ($\mathbf{d}^p$) dans la structure en couches du motif diffractant.

On a ainsi, lors du passage de l'interface en $z = p$, la matrice d'interface $\mathbf{s}^p$ suivante :

$$\begin{bmatrix} u^p(z^p) \\ d^{p+1}(z^p) \end{bmatrix} = \mathbf{s}^p \begin{bmatrix} d^p(z^p) \\ u^{p+1}(z^p) \end{bmatrix}$$

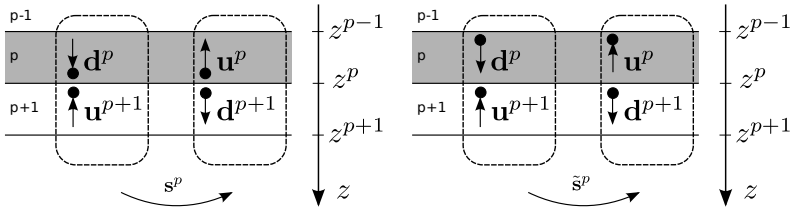
et lors du passage de la couche p en entier, la matrice de couche $\tilde{\mathbf{s}}^p$:

$$\begin{bmatrix} u^p(z^{p-1}) \\ d^{p+1}(z^p) \end{bmatrix} = \tilde{\mathbf{s}}^p \begin{bmatrix} d^p(z^{p-1}) \\ u^{p+1}(z^p) \end{bmatrix} \quad \text{(B.37)}$$

qui vaut :

$$\tilde{\mathbf{s}}^p = \begin{bmatrix} I & 0 \\ 0 & e^{r_m^p(z^{p-1}-z^p)} \end{bmatrix} \mathbf{s}^p \begin{bmatrix} e^{r_m^p(z^{p-1}-z^p)} & 0 \\ 0 & I \end{bmatrix}$$

Les schémas de la figure B.7 représentent les actions des matrices d'interface et de couche dans le cas général.



(a) $\mathbf{s}^p$ (b) $\tilde{\mathbf{s}}_p$

Fig. B.7: Action des opérateurs entre les modes entrants et sortants

On définit la **matrice de pile $\mathbf{S}_p$** comme la matrice liant les modes entrant dans les p premières couches du motif aux modes sortant de ces même couches (cf. fig. B.8).

$$\begin{bmatrix} u^0(0) \\ d^{p+1}(z^p) \end{bmatrix} = \tilde{\mathbf{s}}^p \begin{bmatrix} d^0(0) \\ u^{p+1}(z^p) \end{bmatrix} \quad \text{(B.38)}$$

Naturellement, la solution de notre problème est la matrice, $\mathbf{S}_N$, qui relie l'onde incidente aux multiples modes réfléchis et transmis par le réseau tout entier ($R = u^0$, $\delta_0 = d^0$) :

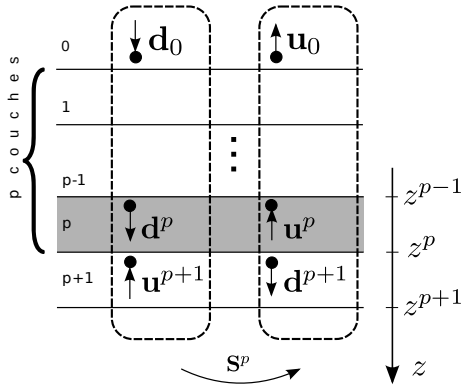


Fig. B.8: Action de la matrice $\mathbf{S}^p$ sur les p premières couches du réseau

$$\begin{bmatrix} R \\ T \end{bmatrix} = \mathbf{S}^N \begin{bmatrix} \delta_0 \\ 0 \end{bmatrix} \tag{B.39}$$

Cette matrice se calcule par récurrence :
- $\mathbf{S}^0 = \mathbf{s}^0$: l'initialisation se fait par la première matrice d'interface ;
- ensuite, pour tout $p \leq N$, on calcule $\mathbf{S}^p$ à l'aide de $\mathbf{S}^{p-1}$ (matrice de pile des $(p-1)$ premières couches) puis de la p-ième matrice de couche $\tilde{\mathbf{s}}^p$. La formule de récurrence est :

$$\begin{cases} (\mathbf{S}^p)_{du} = (\mathbf{S}^{p-1})_{du} + (\mathbf{S}^{p-1})_{uu}(1 - (\tilde{\mathbf{s}}^p)_{du}(\mathbf{S}^{p-1})_{ud})^{-1}(\tilde{\mathbf{s}}^p)_{du}(\mathbf{S}^{p-1})_{dd} \\ (\mathbf{S}^p)_{uu} = (\mathbf{S}^{p-1})_{uu}(1 - (\tilde{\mathbf{s}}^p)_{du}(\mathbf{S}^{p-1})_{ud})^{-1}(\tilde{\mathbf{s}}^p)_{uu} \\ (\mathbf{S}^p)_{dd} = (\tilde{\mathbf{s}}^p)_{dd}[(\mathbf{S}^{p-1})_{dd} + (\mathbf{S}^{p-1})_{ud}(1 - (\tilde{\mathbf{s}}^p)_{du}(\mathbf{S}^{p-1})_{ud})^{-1}(\tilde{\mathbf{s}}^p)_{du}(\mathbf{S}^{p-1})_{dd}] \\ (\mathbf{S}^p)_{ud} = (\tilde{\mathbf{s}}^p)_{dd}(\mathbf{S}^{p-1})_{ud}(1 - (\tilde{\mathbf{s}}^p)_{du}(\mathbf{S}^{p-1})_{ud})^{-1}(\tilde{\mathbf{s}}^p)_{uu} + (\tilde{\mathbf{s}}^p)_{ud} \end{cases}$$

Le mécanisme que nous venons de décrire s'appelle l'algorithme des **matrices de pile** ou des matrices S (*stack matrices*). Il en existe d'autres mais celui-ci est le plus adapté à la méthode modale que l'on utilise (cf. Li *et al.* [58]) notamment grâce à l'inversion dans l'équation B.28 (généralisée) des termes en $e^{r_m^p(z^{p-1}-z^p)}$ présents dans la matrice Ω^+. Les exponentiels croissants sont en effet source d'une importante instabilité numérique (cf. [57]).

ANNEXE C

Code MMFE

Le problème direct en scatterométrie consiste à calculer une signature scatterométrique à partir d'un créneau périodique (de période connue) défini par :
- sa forme géométrique
- les indices optiques des matériaux le constituant

Un nouveau code de calcul électromagnétique basé sur la MMFE [1] a été développé pour servir de base à tout l'édifice *libScattero* (cf. annexe D) en partant de l'implémentation déjà réalisée au laboratoire en langage Matlab. *libMMFE* est la bibliothèque logicielle qui intègre ce code électromagnétique.

Les caractéristiques qui ont justifié cette nouvelle implémentation sont multiples :

1. **code minimaliste et robuste** : souvent destiné à être exécuté un très grand nombre de fois, aucune concession n'est faite quant à la rapidité d'exécution et à la quantité de mémoire utilisée.
2. **code totalement documenté** : l'objectif est de faire de *libMMFE* une bibliothèque standard d'intérêt scientifique et pédagogique.
3. **code portable** : le langage utilisé est le Fortran 77, pour lequel de nombreux compilateurs de très grande qualité sont disponibles sur le marché, et les seules bibliothèques externes utilisées sont les très populaires et très éprouvées BLAS et LAPACK[2].
4. **code immédiatement utilisable** : des interfaces python et C-Ansi sont disponibles.

C.1 Entrées et sorties

La géométrie du système diffractant est modélisée par une succession de couches d'épaisseurs variables. Chaque couche peut être homogène si elle comporte un seul matériau (air, substrat de silicium, etc.) ou inhomogène pour les couches décrivant le motif périodique. La figure C.1 représente

[1] *Modal Method by Fourier Expansion*, Méthode modale par développement de Fourier, voir 3.3

[2] BLAS (Basic Linear Algebra Subprograms [49]) et LAPACK(Linear Algebra PACKage [12]) sont deux bibliothèques logicielles codées en Fortran 77 et destinées aux calculs d'algèbre linéaire : BLAS inclut les opérations de base (addition, multiplication de matrices, etc.) sur lesquelles se base LAPACK pour résoudre un grand nombre de problèmes linéaires (inversion de matrices, problèmes aux valeurs propres).

une telle modélisation : ici 5 couches décrivent le réseau.

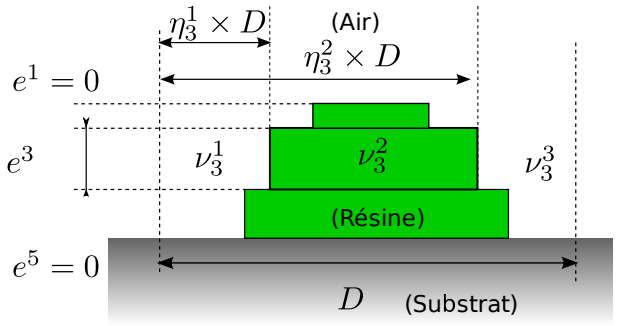


Fig. C.1: Exemple de modélisation d'un créneau simple de résine

Les valeurs à définir pour décrire la géométrie d'un réseau de N couches sont :
- la période D du réseau ;
- les épaisseurs $(e_i)_{1\leq i\leq N}$ de chaque couche. La première et la dernière, homogènes d'épaisseur infinies valent par convention $e_1 = e_N = 0$;
- our chaque couche i, les abscisses normalisées à D (donc valant entre 0 et 1) des changements de matériaux $(\eta_i^j)_{1\leq j\leq (P-1)}$. P est le nombre maximal de matériaux successifs dans le réseau ; dans l'exemple précédent il est de 3 (pour une couche du réseau : Air-Résine-Air).

En plus de cette géométrie, il est nécessaire de définir les conditions de la simulation électromagnétique et du calcul :
- angle d'incidence du rayon lumineux θ ;
- longueur d'onde λ de ce rayon et donc la valeur des indices optiques des matériaux à cette longueur d'onde : ceci sera symbolisé, pour une couche i, par les nombres complexes $(\nu_i^j)_{1\leq j\leq P}$. Dans le schéma précédent, ν_3^1 est l'indice optique de l'air, et ν_3^2 est celui de la résine.
- ordre de troncature du développement en série de Fourier M (cf. 3.3).

En résumé, l'utilisation du code MMFE implique l'entrée des données suivantes :

D	$\mathbb{R}$	période du réseau
N	$\mathbb{N}$	nombre de couches
P	$\mathbb{N}$	nombre de matériaux par couche
$\mathbf{e}$	$\mathbb{R}^N$	épaisseurs des couches
η	$\mathbb{R}^{(P-1)\times N}$	abscisse des changements de matériaux
ν	$\mathbb{C}^{(P-1)\times N}$	indices optiques des matériaux
M	$\mathbb{N}$	ordre de troncature

En plus de cela, la fonction appelante devra spécifier le type de signal ellipsométrique parmi :

$$(S_1, S_2), (\psi, \Delta), (tan(\psi), cos(\Delta)), (I_s, I_c), r_p/r_s$$

et le type de l'ellipsomètre dont on prendra les conventions de calcul (Jobin-Yvon ou KLA/Tencor). Plus de détails sur ces signaux et conventions se trouvent au chapitre 2.

C.2 Modules

Cette liste des modules de la bibliothèque est destinée à faire le lien entre les pages de cet ouvrage traitant de la simulation électromagnétique et l'implémentation du code :

module	rôle	référence
`cren.f`	Développement en série de Fourier de la permittivité périodique d'une couche	équation (B.13)
`mats.f`	Calcul de la matrice S finale	équation (B.39)
`casc1.f`	Cascadage des matrices de couche	équation (B.38)
`tri_1d.f`	Tri des valeurs propres	équation (B.20)
`awad_te.f`	Diagonalisation de $\mathcal{M}$ et tri, mode TE	équations (B.17) et (B.20)
`awad_tm.f`	Diagonalisation de $\mathcal{M}$ et tri, mode TM	
`efficiency.f`	Calcul des efficacités	équations (B.4) et (B.5)
`mmfe_compute.f`	Calcul d'un couple de signaux	chapitre 2
`mmfe_compute_sig.f`	Calcul d'une signature	

C.3 Exemples d'utilisation

Nous simulons ici la réponse d'un créneau constitué d'un diélectrique d'indice $\tilde{n} = 0.22 - i6.71$. Ce créneau "canonique" (déjà mentionné dans [62] et [33]) est représenté sur la figure C.2 : il a une période et une profondeur de $1\mu m$ et est illuminé sous un angle de 30°par une longueur d'onde $\lambda = 1\mu m$.

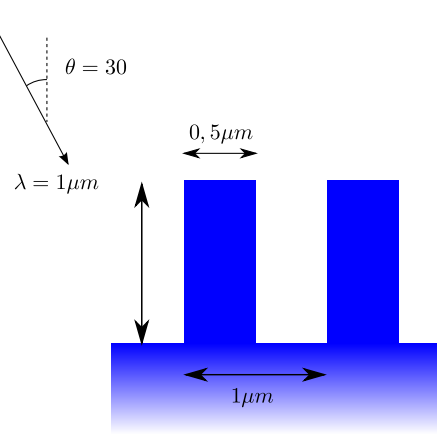


Fig. C.2: Simulation de la réponse électromagnétique d'un réseau simple de diélectrique

Ici, $N = 3$, $P = 3$ et les tableaux sont choisis ainsi :

$$e = \begin{bmatrix} 0 \\ 1000 \\ 0 \end{bmatrix} \quad \eta = \begin{bmatrix} 0 & 1 \\ 0.25 & 0.75 \\ 0 & 1 \end{bmatrix} \quad \nu = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \tilde{n} & 1 \\ \tilde{n} & \tilde{n} & \tilde{n} \end{bmatrix}$$

Les listings C.1 et C.2 sont deux exemples de codes minimaux (respectivement en langage C++ et Fortran 77) faisant appel à la bibliothèque *libMMFE* pour calculer les signaux I_S et I_C correspondant au problème.

```cpp
#include "mmfe.h"

int main(void)
{
        double d1, d2;

        complex air = complex(1,0),
                mat = complex(0.22,-6.71);


        // Definition du reseau

        double d = 1000;
        int n = 3, p = 3;

        double ec[3] = { 0, 1000, 0 };

        double eta[6]=
                { 0,    1,
                  0.25, 0.75,
                  0,    1    };

        complex nu_lam[9] = {
                air, air, air,
                air, mat, air,
                mat, mat, mat
        };

        double teta = 30;
        double lambda = 1000;

        int sigtype = 6,    // ls/lc
                ellipso = 1,// Jobin-Yvon
                M=20 ;

        // Appel a libMMFE

        mmfe_compute_( &d,
                       ec, eta, nu_lam,
                       &n, &p,
                       &teta, &lambda,
                       &M,
                       &sigtype, &ellipso,
                       &d1, &d2);

        printf("%f %f\n", d1, d2);









}
```

Listing C.1: Exemple de code C++

```fortran
      PROGRAM TEST_MMFE

      IMPLICIT NONE

      INTEGER M, N, P, SIGTYPE, ELLIPSO
      DOUBLE PRECISION D,EC( 3 ),  ETA( 2, 3 ),
     $     TETA, LAMBDA0

      DOUBLE COMPLEX  NU_LAM( 3, 3 )
      DOUBLE PRECISION D1, D2

      DOUBLE COMPLEX MAT, AIR
      PARAMETER (MAT = (0.22, -6.71))
      PARAMETER (AIR = (1, 0))

C     DEFINITION DU RESEAU

      D = 1000
      N = 3
      P = 3

      DATA EC /0, 1000, 0/

      DATA ETA
     $     / 0,    1,
     $       0.25, 0.75,
     $       0,    1    /

      DATA NU_LAM
     $     / AIR, AIR, AIR,
     $       AIR, MAT, AIR,
     $       MAT, MAT, MAT /


C     DEFINITION DES CONDITION DE CALCUL

      LAMBDA0 = 1000
      TETA = 30

      SIGTYPE = 6
      ELLIPSO = 1
      M=20

c     APPEL A LIBMMFE

      CALL MMFE_COMPUTE( D, EC, ETA, NU_LAM, N, P,
     $     TETA, LAMBDA0, M, SIGTYPE, ELLIPSO,
     $     D1, D2)

      PRINT *, D1, D2


      END
```

Listing C.2: Exemple de code Fortran

La sortie est : `0.219359 0.436054`

Calcul de signatures. Il est possible de faire varier dans une boucle la valeur de λ et, à condition de mettre à jour le tableau ν, de calculer une signature complète. La fonction `mmfe_compute_sig.f` sert à cela et possède sensiblement le même prototype que `mmfe_compute.f` à l'exception que `lambda` est un tableau 1D et `nu_lam` un tableau 3D (la dimension supplémentaire est celle des différentes longueurs d'ondes).

ANNEXE D

libScattero

Les travaux de cette thèse ont dès le début nécessité un effort important de développement informatique. Il nous a par conséquent semblé indispensable de constituer très tôt une architecture logicielle cohérente afin d'organiser toutes les routines et structures de données qui ont été développées. Cette structure, *libScattero* , a pris la forme d'une bibliothèque logicielle orientée-objet destinée à la scatterométrie en général, statique comme dynamique.

D.1 Structure

Dans le vocabulaire de la programmation orientée-objet, une **classe** représente une sorte de programme moule avec lequel sont fabriqués des **objets** : elle définit notamment des attributs (données) et des méthodes (actions, comportements). Une classe **hérite** d'une autre si elle la spécialise, c'est-à-dire si elle possède, en plus des propriétés de la classe-mère, un ensemble d'attributs et de méthodes qui la différencie[1].

libScattero est un ensemble de 7 classes C++ décrites dans le tableau D.1 et organisées comme cela est représenté sur le diagramme D.1 : le programmeur peut instancier une de ces classes et ainsi créer des objets qui seront les représentations en mémoire d'une signature scatterométrique, d'un film de signatures ou encore d'un créneau diffractant.

L'intérêt de cette structure orientée-objet est double :

- D'une part, elle procure une plus grande facilité d'utilisation au programmeur utilisateur de la bibliothèque : il peut manipuler les objets, les faire interagir (on résout un problème inverse en faisant simplement interagir une signature avec une bibliothèque) sans se préoccuper du stockage des données, des algorithmes mis en jeux, etc.
- D'autre part, pour le programmeur développeur de *libScattero* , la modularité de cette structure rend la bibliothèque évolutive. L'ajout d'une fonctionnalité se fait simplement par l'ajout d'une méthode qui peut interagir avec les autres ; l'architecture globale ne change pas.

[1] Cette description trop succincte ne saurait expliquer la puissance et la subtilité des tous les concepts de la programmation orientée-objet. Le lecteur intéressé pourra se reporter sur les ouvrages consacrés [70].

`lSObject`	Classe principale dont héritent toutes les autres à des degrés divers. Elle assure entre autre la cohérence de l'architecture et les liens entre objets.
`lSMaterial`	Classe représentant un matériau, et notamment ses indices optiques.
`lSGrating`	Classe représentant un créneau de résine modélisé (il comporte donc une liste d'objets `lSMaterial`)
`lSSigSet`	Classe représentant un jeu de signature scatterométrique.
`lSSignature`	Classe représentant une signature scatterométrique individuelle.
`lSSigMovie`	Classe héritant de `lSSigSet` ; c'est un jeu de signatures indexées par le temps : un film de signatures
`lSSigLib`	Classe héritant de `lSSigSet` ; c'est un jeu de signatures indexées par un jeu de paramètres géométriques : une bibliothèque.

Tab. D.1: Description sommaire des classes de *libScattero*

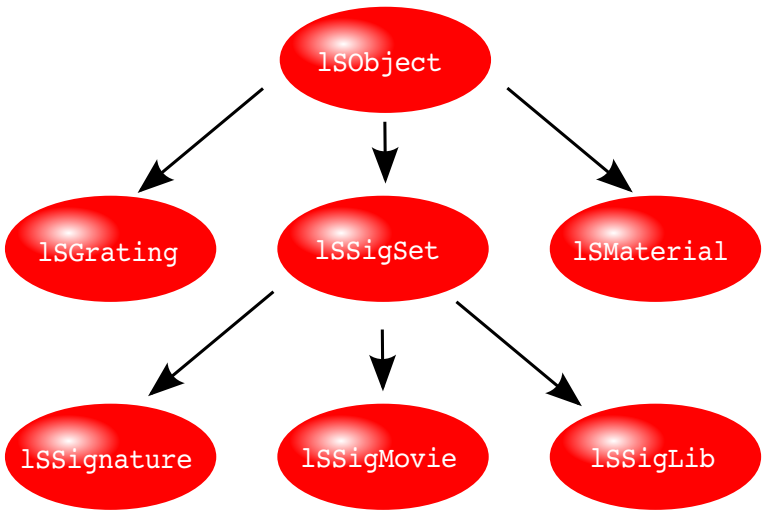


Fig. D.1: Architecture orientée-objet de *libScattero*

D.2 Fonctionnalités

Nous allons ici décrire, classe par classe, les fonctionnalités offertes par *libScattero* . L'intégralité du code constitue environ 10 000 lignes et intègre tout ce qui a été développé pour ces travaux de thèse : nouveaux algorithmes comme routines communes.

Classe	Fonctionnalité
`lSObject`	Aucune fonctionnalité directement utilisable.
`lSMaterial`	Ouverture/export des fichiers d'indices en différents formats (texte, XML). Gestion des dépendances en longueur d'onde (ou énergies) et en température. Interpolation automatique des indices.
`lSGrating`	Construction des formes géométriques par fichiers script `.gra`, (cf. D.3) Dessin de la forme en format d'image vectoriel (`.svg`) **Calcul d'une signature ou d'une bibliothèque** (appel au code MMFE) sur plusieurs processeurs (MPI)
`lSSigSet`	Manipulation du spectre des longueurs d'ondes (réduction, extension) Manipulation de la base de signatures : addition de jeux, suppression Conversion : eV $\leftrightarrow$ nm Conversion : entre types de signaux (Ψ, Δ), (I_S, I_C), etc. Bruitage de signatures. Extraction d'une signature sous forme d'objet `lSSignature`
`lSSignature`	Ouverture des fichiers signature Jobin-Yvon (format `.spe`) Dessin (format `.eps`)
`lSSigMovie`	Ouverture des fichiers films Jobin-Yvon (format `.kin` et collection de `.spe`) **Comportement dynamique** : l'objet accroît sa taille en fonction des signatures acquises en temps réel **Communication avec les ellipsomètres Jobin-Yvon** : acquisition des mesures en temps réel par liaison TCP/IP Montage : découpage et assemblage de morceaux de films. Filtrage : atténuation du bruit de certains canaux. **Reconstruction (temps réel ou a posteriori) de l'évolution d'un profil diffractant** Export du film d'un profil qui varie (format vidéo `.avi`).
`lSSigLib`	**Problème inverse : méthode des bibliothèques** Utilisation des processeur graphiques NVIDIA pour les cas critiques (bibliothèque de grande taille et fréquence d'acquisition élevée)

D'autres caractéristiques destinées à faciliter l'usage de *libScattero* doivent être mentionnées :

- Interface avec le langage Python : La souplesse de ce langage permet à *libScattero* d'être utilisée aussi bien en ligne que dans des scripts faciles à écrire.
- Documentation : Une bibliothèque logicielle n'est utilisable par le programmeur que si chacun de ses éléments est rigoureusement documenté. Cette documentation est générée automatiquement dans les formats HTML (pages web) et PDF par l'outil Doxygen.

D.3 Construction d'un modèle géométrique

La base de la scatterométrie est l'établissement d'un modèle géométrique paramétré. Les mesures faites permettent ensuite de déterminer ces paramètres.

La classe `lSGrating` est conçue pour cette modélisation. Elle dispose notamment des méthodes pour interpréter un langage de script dédié. Un exemple d'un tel script est donné sur le listing D.1 ; il correspond au créneau de résine modélisé représenté sur la figure D.2.

```
1  MAT          CST        1          0
2  MAT          FILE       indices_n_k_M784.txt
3  MAT          FILE       Si50.txt
4  MAT          FILE       SiO2_HJY.ref.txt
5
6  SET          H          2          2          600
7  SET          CD         2          2          600
8  SET          ANGLE      15
9  SET          E_OX       1.5
10
11 INIT         10         1000       CD         H
12 SLOPE        ANGLE
13 ADD_LA       9          E_OX
14 FILL_LAY     9          3
```

Listing D.1: Exemple de fichier script .gra

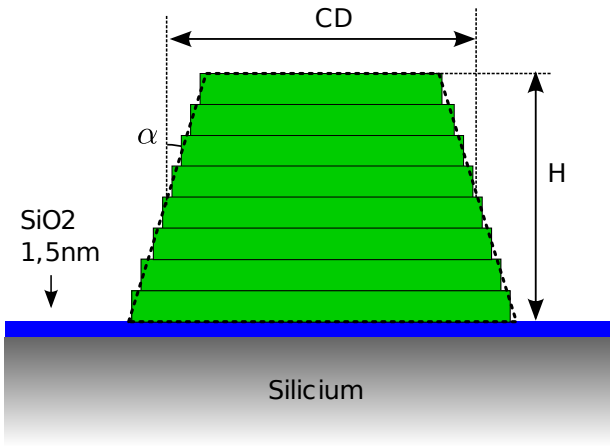


Fig. D.2: Exemple de créneau de résine modélisé en utilisant le fichier script D.1 : un créneau carré (largeur CD et hauteur H) de 10 couches (air - 8 couches de résine - substrat) est d'abord créé. Un angle de pente α est ajouté. Une sous-couche de 15$\mathring{A}$ est créée, puis "remplie" par de l'oxyde de silicium

Dans chaque ligne du script, figure en première colonne l'instruction, et dans les suivantes, les paramètres. Un paramètre peut être une valeur entrée directement (comme les deux premiers de l'instruction INIT, ligne 11), un symbole vers une valeur ou un ensemble de valeurs (deux derniers de INIT), ou un mot clé.

- On déclare un matériau avec l'instruction MAT, suivie du mot clé CST si l'indice est constant, ou FILE s'il est défini par un fichier de données. Sur l'exemple D.1, l'air, d'indice constant est défini en premier, suivi des matériaux résine, silicium et oxyde de silicium.
- On déclare une valeur ou un ensemble de valeurs par SET. La ligne 6 signifie que H représente des valeurs allant de 2 à 600 par pas de 2. La ligne 9 signifie que l'épaisseur d'oxyde SiO_2 est constante et vaut $1,5nm$.
- On définit un modèle géométrique avec les instructions de construction : INIT est la première ; elle crée simplement un réseau carré (de période 1000, de hauteur H, de largeur CD et subdivisé en 10 couches dans l'exemple). Ensuite, on peut utiliser d'autres instructions qui permettent

de manipuler la géométrie de ce réseau. La liste de ces instructions est facilement extensible ; nous en avons compilé quelques-unes dans le tableau D.2.

mnémonique	action
SLOPE	Crée une pente sur les flancs du créneau.
ADD_LA	Ajoute une couche.
FILL_LAY	Redéfinit le matériau d'une couche.
RNDTOP	Arrondit les coins du haut du créneau.
RNDFOOT	Arrondit les coins aux pieds du créneau.
FLU	Fluage du créneau (cf. chapitre 8).

Tab. D.2: Quelques instructions de dessin

Ces fichiers script permettent à eux seuls de générer une bibliothèque : pour chaque combinaison des paramètres définis par les instructions SET, une signature est calculée.

D.4 Langages et bibliothèques utilisées

Le fonctionnement de *libScattero* n'est pas autonome ; il se base sur des bibliothèques logicielles indépendantes. Pour une utilisation pleinement fonctionnelle de *libScattero* sur un système, chacune de ces bibliothèques doit être disponible.

- **libMMFE** : Il s'agit du code de calcul électromagnétique responsable de la génération de signatures et de bibliothèques. L'annexe C lui est dédiée.
- **BLAS/LAPACK** : liée au fonctionnement de *libMMFE*
- **MPI** : *Message Passing Interface*, il s'agit à proprement parler d'une spécification de bibliothèque qui rend possible un calcul entre plusieurs processeurs en organisant les communications (par messages) entre eux. Il existe de nombreuses implémentations de MPI faites pour des architectures différentes (processeurs à mémoire partagée, grappes de calcul). Nous avons utilisé l'implémentation libre MPICH ([34], [35]) pour effectuer nos calculs sur 4 unités à mémoire partagée : deux processeurs de deux coeurs.
- **Cg/OpenGL** : Ces bibliothèques ne sont pas indispensables mais permettent l'utilisation du processeur graphique (cf. chapitre 6)

D.5 Outils

La bibliothèque *libScattero* inclue un certain nombre d'outils destinés à la rendre utilisable directement par l'utilisateur.

D.5.1 IS/lscl

`lscl` est un outil en ligne de commande pour accéder à toutes les fonctions et algorithmes disponibles (problème direct, inverse, reconstruction etc.). Cet outil est stable et a été utilisé très couramment durant cette thèse.

D.5.2 IS/indices

Cet outil utilise *libScattero* de manière détournée pour la détermination des courbes de dispersion des matériaux (voir à ce sujet la partie IV) en se basant sur la méthode des bibliothèques. L'utilisation

nécessite un fichier de données *signature* ou *film* ellipsométrique du matériau qui a été couché sur substrat de silicium, ainsi que l'épaisseur de cette couche.

Bibliographie

[1] C for graphics home page. http://developer.nvidia.com/page/cg_main.html.

[2] Cuda for gpu computing. http://news.developer.nvidia.com/2007/02/cuda_for_gpu_co.html.

[3] Gnu project - autoconf. http://www.gnu.org/software/autoconf/.

[4] Gnu project - automake. http://www.gnu.org/software/automake.

[5] The mesa 3d graphics library. http://www.mesa3d.org/.

[6] Mpich-a portable implementation of mpi. http://www-unix.mcs.anl.gov/mpi/mpich1/index.htm.

[7] Nvidia geforce 8800 gtx. http://www.nvidia.com/page/geforce_8800.html.

[8] Opengl, the industry's foundation for high performance graphics. http://www.opengl.org/.

[9] Feuille de route itrs 2007. http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_Metrology.pdf, 2007.

[10] O. Acher, E. Bigan, and B. Drévillon. Improvements of phase-modulated ellipsometry. *Review of Scientific Instruments*, 60(1) :65–77, 1989.

[11] ACM, editor. *Hardware acceleration for spatial selections and joins*, 2003.

[12] E. Angerson, Z. Bai, J. Dongarra, A. Greenbaum, A. Mckenney, J. Du Croz, S. Hammarling, J. Demmel, C. Bischof, C. Bischof, D. Sorensen, and A10. Lapack : A portable linear algebra library for high-performance computers. In *Supercomputing '90. Proceedings of*, page 2–11, 1990.

[13] Chas N. Archie, editor. *Real-time profile shape reconstruction using dynamic scatterometry*, volume 6518. SPIE, 2007.

[14] Robert G. Belleman, Jeroen Bédorf, and Portegies. High performance direct gravitational n-body simulations on graphics processing units ii : An implementation in cuda. *New Astronomy*, 13(2) :103–112, February 2008.

[15] Oliver Deussen Stefan Hiller Benjamin Bustos and Daniel Keim. A graphics hardware accelerated algorithm for nearest neighbor search. In Geert Dick van Albada Peter M.A. Sloot Vassil N. Alexandrov and Jack Dongarra, editors, *Computational Science – ICCS 2006*, volume 3994 of *LNCS*, page 196–199. Springer, 2006.

[16] Franck Bernoux. Ellipsométrie - théorie. *Techniques de l'Ingénieur*, Mesures et contrôle R 6490, 2003.

[17] F. Bloch. Über die Quantenmechanik der Elektronen in Kristallgittern. 52 :555–600, 1928.

[18] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2008.

[19] Ian Buck, Naga Govindaraju, Mark Harris, Jens Kruger, Aaron Lefohn, David Luebke, Tim Purcell, and Cliff Woolley. Gpgpu : General-purpose computation on graphics hardware. In *Course 32 at ACM SIGGRAPH*, 2004.

[20] J. Chandezon, D. Maystre, and G. Raoult. A new theoretical method for diffraction gratings and its numerical application. *J.Opt.-Nouv. Rev. d'Opt.*, 11(4) :235–241, 1980.

[21] Stephen Y. Chou, Peter R. Krauss, and Preston J. Renstrom. Nanoimprint lithography. *Journal of vacuum science & technology*, 14(6) :4129–4133, November 1996.

[22] P Craven and G Wahba. Smoothing noisy data with spline functions. *Numer. Math.*, 31 :377–403, 1979.

[23] Xuegong Deng, Lei Chen, Jr. Paul F. Sciortino, Feng Liu, and Jian J. Wang. Nondestructive metrology for nanoimprint processes. *Journal of Vacuum Science and Technology B : Microelectronics and Nanometer Structures*, 24(2) :686–689, 2006.

[24] G. Floquet. Sur les équations différentielles linéaires à coefficients périodiques. *Annales scientifiques de l'École Normale Supérieure*, Sér. 2, 12 :47–88, 1883.

[25] David Fuard, Corinne Perret, Vincent Farys, Cecile Gourgon, and Patrick Schiavone. Measurement of residual thickness using scatterometry. *Journal of vacuum science & technology*, 23(6) :3069–3074, 2005.

[26] C.G. Galarza, P.P. Khargonekar, and F.L. Jr. Terry. Real-time estimation of patterned wafer parameters using in situ spectroscopic ellipsometry. *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, 1 :773–778 vol. 1, 1999.

[27] Vincent Garcia, Eric Debreuve, and Michel Barlaud. Fast k nearest neighbor search using gpu. *ArXiv*, Apr 2008.

[28] T. K. Gaylord and M. G. Moharam. Analysis and applications of optical diffraction by gratings. *Proceedings of the IEEE*, 73(5) :894–937, 1985.

[29] T.K. Gaylord and M.G. Moharam. Analysis and applications of optical diffraction by gratings. *Proceedings of the IEEE*, 73(5) :894–937, May 1985.

[30] Issam Gereige, Stéphane Robert, Sylvie Thiria, Fouad Badran, Gérard Granet, and Jean Jacques Rousseau. Recognition of diffraction-grating profile using a neural network classifier in optical scatterometry. *J. Opt. Soc. Am. A*, 25(7) :1661–1667, 2008.

[31] Naga K. Govindaraju, Brandon Lloyd, Wei Wang, Ming Lin, and Dinesh Manocha. Fast computation of database operations using graphics processors. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Courses*, page 206, New York, NY, USA, 2005. ACM.

[32] G Granet. Analysis of diffraction by crossed gratings using a non-orthogonal coordinate system. *Pure and Applied Optics : Journal of the European Optical Society Part A*, 4(6) :777–793, 1995.

[33] G. Granet and B. Guizal. Efficient implementation of the coupled-wave method for metallic lamellar gratings in tm polarization. *J. Opt. Soc. Am. A*, 13(5) :1019, 1996.

[34] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing*, 22(6) :789–828, sep 1996.

[35] William D. Gropp and Ewing Lusk. *User's Guide for* `mpich`*, a Portable Implementation of MPI*. Mathematics and Computer Science Division, Argonne National Laboratory. ANL-96/6.

[36] Marc Hamdorf and Diethelm Johannsmann. Erratum : "surface-rheological measurements on glass forming polymers based on the surface tension driven decay of imprinted corrugation gratings" [j. chem. phys. [bold 112], 4262 (2000)]. *The Journal of Chemical Physics*, 114(21) :9685–9685, 2001.

[37] Jerome Hazart, Gilles Grand, Philippe Thony, David Herisson, Stephanie Garcia, and Oliver Lartigue. Spectroscopic ellipsometric scatterometry : sources of errors in critical dimension control. *SPIE Process and Materials Characterization and Diagnostics in IC Manufacturing*, 5041(1) :9–20, 2003.

[38] H.Kawahira. Changes of chemical nature of photoresists induced by various plasma treatments and their impact on lwr. In *Proc. SPIE 6153*, 2006.

[39] James M. Holden, Thomas Gubiotti, William A. McGahan, Mircea V. Dusa, and Ton Kiers. Normal-incidence spectroscopic ellipsometry and polarized reflectometry for measurement and control of photoresist critical dimension. In Daniel J. C. Herr, editor, *Metrology, Inspection, and Process Control for Microlithography XVI*, volume 4689, page 1110–1121. SPIE, 2002.

[40] J. H. Holland. *Adaptation In Natural And Artificial Systems*. University of Michigan Press, 1975.

[41] F. Hua, Y. Sun, A. Gaur, M.A. Meitl, L. Bilhaut, L. Rotkina, J. Wang, P. Geil, M. Shim, J.A. Rogers, and A. Shim. Polymer imprint lithography with molecular-scale resolution. *Nano Letters*, 4(12) :2467–2471, 2004.

[42] Hsu-Ting Huang and Jr. Fred L. Terry. Spectroscopic ellipsometry and reflectometry from gratings (scatterometry) for critical dimension measurement and in situ, real-time process monitoring. *Thin Solid Films*, 455-456 :828–836, 2004.

[43] P.J. Huber. *Robust Statistics*. John Wiley, 1981.

[44] T. Jannson and N. C. Gallagher, editors. *Subwavelength photoresist grating metrology using scatterometry*, volume 2532, sep 1995.

[45] P.R. Johnston and R.M. Gulrajani. Selecting the corner in the l-curve approach to tikhonov regularization. *Biomedical Engineering, IEEE Transactions on*, 47(9) :1293–1296, Sept. 2000.

[46] H.A. Kramers. La diffusion de la lumiere par les atomes. *Transactions of Volta Centenary Congress*, (2) :545–557, 1927.

[47] R. D. L. Kronig. On the theory of dispersion of x'rays. *Journal of the Optical Society of America (1917-1983)*, 12 :547–+, nov 1926.

[48] Philippe Lalanne and G. Michael Morris. Highly improved convergence of the coupled-wave method for tm polarization. *J. Opt. Soc. Am. A*, 13(4) :779, 1996.

[49] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Algorithm 539 : Basic Linear Algebra Subprograms for Fortran usage [F1]. *ACM Transactions on Mathematical Software*, 5(3) :324–325, September 1979. See also [**?**, **?**, **?**, **?**].

[50] T. Leveder, S. Landis, and L. Davoust. Imprint time optimization in hot embossing lithography. In *Emerging Lithographic Technologies XI. Proceedings of the SPIE.*, 2007.

[51] T. Leveder, S. Landis, and L. Davoust. Reflow dynamics of thin patterned viscous films. *Applied Physics Letters*, 92(1) :013107, 2008.

[52] T. Leveder, S. Landis, L. Davoust, and N. Chaix. Optimization of demolding temperature for throughput improvement of nanoimprint lithography. *Microelectron. Eng.*, 84 :953–957, 2007.

[53] T. Leveder, S. Landis, L. Davoust, S. Soulan, J.-H. Tortai, and N. Chaix. Surface characterization of imprinted resist above glass transition temperature. *Journal of Vacuum Science and Technology*, 25(6) :2365–2369, 2007.

[54] Tanguy Leveder. *Etude et caractérisation de films nanométriques de polymère. Application a la lithographie par nanoimpression*. PhD thesis, Institut National Polytechnique de Grenoble, 2009.

[55] Tanguy Leveder, Stefan Landis, Laurent Davoust, Sebastien Soulan, and Nicolas Chaix. Demolding strategy to improve the hot embossing throughput. *SPIE Emerging Lithographic Technologies XI*, 6517(1) :65170N, 2007.

[56] Harry J. Levinson. *Principles of Lithography*. SPIE - The International Society for Optical Engineering, 2005.

[57] Lifeng Li. Multilayer modal method for diffraction gratings of arbitrary profile, depth, and permittivity. *J. Opt. Soc. Am. A*, 10(12) :2581, 1993.

[58] Lifeng Li. Formulation and comparison of two recursive matrix algorithms for modeling layered diffraction gratings. *J. Opt. Soc. Am. A*, 13(5) :1024, 1996.

[59] Lifeng Li. Use of fourier series in the analysis of discontinuous periodic structures. *J. Opt. Soc. Am. A*, 13(9) :1870–1876, 1996.

[60] Lifeng Li. New formulation of the fourier modal method for crossed surface-relief gratings. *J. Opt. Soc. Am. A*, 14(10) :2758–2767, 1997.

[61] Lifeng Li, Jean Chandezon, Gérard Granet, and Jean-Pierre Plumey. Rigorous and efficient grating-analysis method made easy for optical engineers. *Appl. Opt.*, 38(2) :304–313, 1999.

[62] Lifeng Li and Charles W. Haggans. Convergence of the coupled-wave method for metallic lamellar diffraction gratings. *J. Opt. Soc. Am. A*, 10(6) :1184–1189, 1993.

[63] Lifeng Li and Charles W. Haggans. Convergence of the coupled-wave method for metallic lamellar diffraction gratings. *J. Opt. Soc. Am. A*, 10(6) :1184, 1993.

[64] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, page 91–110, 2003.

[65] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.

[66] William R. Mark, R. Steven Glanville, Kurt Akeley, and Mark J. Kilgard. Cg : a system for programming graphics hardware in a c-like language. *ACM Trans. Graph.*, 22(3) :896–907, 2003.

[67] David Marshall. Nearest neighbour searching in high dimensional metric space. Master's thesis, Australian National University - Department of Computer Science, 2006.

[68] Mateo and Kazuhiro Otsuka. Real-time visual tracker by stream processing. *Journal of Signal Processing Systems*, 2008.

[69] H. L. Maynard, N. Layadi, and J. Tseng-Chung Lee. Multiwavelength ellipsometry for real-time process control of the plasma etching of patterned samples. *Journal of Vacuum Science and Technology B : Microelectronics and Nanometer Structures, Volume 15, Issue 1, January 1997, pp.109-115*, 15 :109–115, jan 1997.

[70] Bertrand Meyer. *Conception et programmation orientées objet*. Eyrolles, 2000.

[71] NewAuthor0. Gpgpu : :reduction tutorial. `http://www.mathematik.uni-dortmund.de/~goeddeke/gpgpu/tutorial2.html`.

[72] Xinhui Niu, N. Jakatdar, Junwei Bao, and C.J. Spanos. Specular spectroscopic scatterometry. *Semiconductor Manufacturing, IEEE Transactions on*, 14(2) :97–111, May 2001.

[73] Jon L. Opsal, Hanyou Chu, and Jingmin Leng. Finite difference algorithm in real-time optical cd applications. *SPIE Metrology, Inspection, and Process Control for Microlithography XVIII*, 5375(1) :1356–1363, 2004.

[74] Jon L. Opsal, Hanyou Chu, Youxian Wen, Yia-Chung Chang, and Guangwei Li. Fundamental solutions for real-time optical cd metrology. In Daniel J. C. Herr, editor, *SPIE Metrology, Inspection, and Process Control for Microlithography XVI*, volume 4689, page 163–176. SPIE, 2002.

[75] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krüger, Aaron E. Lefohn, and Timothy J. Purcell. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1) :80–113, 2007.

[76] Feynman Richard P., Leighton Robert B., and Sands Matthew. *Le cours de physique de Feynman : électromagnétisme*, volume 2. 1979.

[77] D. M. Pai and K. A. Awada. Analysis of dielectric gratings of arbitrary profiles and thicknesses. *J. Opt. Soc. Am. A*, 8(5) :755, 1991.

[78] Erwine Pargon. *Analyse des mécanismes mis en jeu lors de l'élaboration par gravure plasma de structures de dimensions deca-nanométriques : Application au transistor CMOS ultime*. PhD thesis, Université Joseph FOURIER Grenoble, 2004.

[79] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran*. Cambridge University Press, Cambridge, 2 edition, 1992.

[80] Richard Quintanilha. *Etude du problème inverse en diffractométrie spectroscopique*. PhD thesis, Institut national polytechnique de grenoble, 2005.

[81] Benferhat R. Design of new in situ spectroscopic phase modulated ellipsometer. *Le Vide, les couches minces*, 47(258) :264–273, 1991.

[82] A. Rathsfeld, G. C. Hsiao, and J. Elschner. Grating profile reconstruction based on finite elements and optimization techniques. *SIAM Journal on Applied Mathematics*, 64(2) :525–545, 2004.

[83] Lord Rayleigh. On the dynamical theory of gratings. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character (1905-1934)*, 79(532) :399–416, Aug. 1907.

[84] C. J. Raymond, M. E. Littau, A. Chuprin, and S. Ward. Comparison of solutions to the scatterometry inverse problem. In R. M. Silver, editor, *Metrology, Inspection, and Process Control for Microlithography XVIII. Edited by Silver, Richard M. Proceedings of the SPIE, Volume 5375, pp. 564-575 (2004).*, pages 564–575, May 2004.

[85] Christopher J. Raymond, Michael E. Littau, Andrei Chuprin, and Simon Ward. Comparison of solutions to the scatterometry inverse problem. In Richard M. Silver, editor, *SPIE Metrology, Inspection, and Process Control for Microlithography XVIII*, volume 5375, page 564–575. SPIE, 2004.

[86] H. Schift. Nanoimprint lithography : An old story in modern times ? A review. *Journal of Vacuum Science Technology B : Microelectronics and Nanometer Structures*, 26 :458, 2008.

[87] U Schmitt and A K Louis. Efficient algorithms for the regularization of dynamic inverse problems : I. theory. *Inverse Problems*, 18(3) :645–658, 2002.

[88] U Schmitt, A K Louis, C Wolters, and M Vauhkonen. Efficient algorithms for the regularization of dynamic inverse problems : Ii. applications. *Inverse Problems*, 18(3) :659–676, 2002.

[89] Daniel Sjöberg, Christian Engström, Gerhard Kristensson, David J. N. Wall, and Niklas Wellander. A floquet-bloch decomposition of maxwell's equations, applied to homogenization. Technical report, Department of Electroscience Electromagnetic Theory, Lund Institute of Technology, 2003.

[90] Sébastien Soulan, Maxime Besacier, Tanguy Leveder, and Patrick Schiavone. In-line etching process control using dynamic scatterometry. In Harald Bosse, Bernd Bodermann, and Richard M. Silver, editors, *SPIE Modeling Aspects in Optical Metrology*, volume 6617. SPIE, 2007.

[91] Robert Stephens. A survey of stream processing. *Acta Informatica*, 34(7) :491–541, 1997.

[92] AN Tikhonov and VA Arsenin. *Solution of Ill-posed Problems*. 1977.

[93] Pedro Trancoso and Maria Charalambous. Exploring graphics processor performance for general purpose applications. In *DSD '05 : Proceedings of the 8th Euromicro Conference on Digital System Design*, page 306–313, Washington, DC, USA, 2005. IEEE Computer Society.

[94] R. Wood. On a remarkable case of uneven distribution of light in a diffraction grating spectrum. *Phil. Mag*, (4) :396–402, 1902.

[95] Fang Xu and K. Mueller. Accelerating popular tomographic reconstruction algorithms on commodity pc graphics hardware. *Nuclear Science, IEEE Transactions on*, 52(3) :654–663, June 2005.