

Appel à projets FARMAN 2012

# SIMSURF

**Vers une simulation réaliste des états de  
surfaces par calculs massivement parallèles  
sur processeurs graphiques**

Pierre-Alain BOUCARD (LMT, resp. scientifique)

Sylvain LAVERNHE (LURPA)

Hugo LECLERC (LMT)

Christophe TOURNIER (LURPA, resp. scientifique)



## Résumé

Pour diminuer le temps de cycle de développement des produits, le LURPA développe des méthodes afin de simuler l'évolution des écarts géométriques des surfaces tout au long du processus de fabrication. Le développement d'architectures massivement parallèles basées sur des microprocesseurs banalisés permettrait d'augmenter la qualité et la complexité des simulations, mais il nous faut concevoir des logiciels adaptés à ces nouvelles architectures. Le LMT s'est d'ores et déjà engagé dans cette voie afin de réduire les temps de calcul de structures et possède une très bonne expérience dans les calculs hautes performances.

Le projet SIMSURF a donc pour objet d'adapter les logiciels de simulation des états de surfaces obtenus par usinage, pour être exécutés et optimisés sur des architectures parallèles, basées sur des processeurs graphiques. Pour cela, nous nous fixons plusieurs objectifs :

- le découpage intelligent du problème pour tenir compte des forces et faiblesses de ces architectures,
- la visualisation des résultats de simulation sans échange de données avec des logiciels externes,
- la prise en compte de plusieurs échelles de simulation et le passage en temps réel de l'une à l'autre
- La corrélation entre simulation et états de surface réels.

Ce projet, d'une durée de 2 ans, est porté par Christophe Tournier (LURPA, 10 %) et Pierre-Alain Boucard (LMT, 10 %) avec la participation de Sylvain Lavernhe (LURPA, 10 %) et Hugo Leclerc (LMT, 10 %).

La première partie du travail sera menée par un étudiant, Felix Abecassis, de l'école d'ingénieur EPITA. Le travail s'effectuera dans le cadre de son stage de dernière année d'une durée de 6 mois. Il sera employé par l'ENS Cachan en CDD dont le financement sera assuré sur fonds propres pour un montant de 12 k€.

Pour la réussite du projet, nous devons investir dans l'achat d'un nouveau crayon de mesure pour un montant de 9 k€, dans la mise à jour du logiciel de post traitement pour 2 k€ et enfin 3 k€ seront investis dans l'évolution des nœuds graphiques du cluster du LMT.

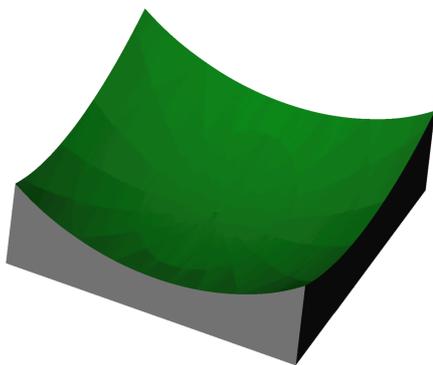
Le coût total du projet est donc de 26 k€ et nous souhaitons obtenir le soutien de l'institut Farman pour un montant de 14 k€.

## Problématique

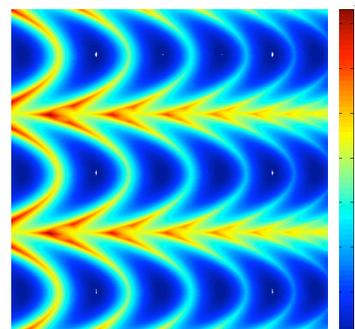
Quelle que soit l'industrie considérée, l'ensemble des produits manufacturés possède des surfaces qui interagissent avec des éléments extérieurs et qui sont conçues pour assurer des fonctions techniques : étanchéité, frottement, propriétés optiques ou d'écoulement hydro ou aéro dynamique. La fonction technique de la surface impose ses écarts géométriques admissibles et donc son processus de fabrication. En particulier, pour diminuer le temps de cycle de développement des produits, il est essentiel de pouvoir simuler l'évolution des écarts géométriques des surfaces tout au long du processus de fabrication.

A l'heure actuelle, la simulation des surfaces usinées dans les logiciels de FAO est très succincte. Il s'agit de simulations géométriques d'outils fictifs représentés par des sphères ou des cylindres sans aucune représentation des arêtes de coupe. De plus, ces simulations n'intègrent aucune caractéristique du processus réel susceptible de détériorer les états de surface lors des opérations d'usinage ou de polissage. Enfin, ces simulations ne permettent pas d'obtenir la précision requise dans des temps raisonnables (<1 min) ni de sélectionner une zone dans laquelle l'utilisateur souhaite avoir davantage de précision. Il existe cependant de nombreuses méthodes de calcul pour effectuer les simulations d'usinage dans la littérature : méthodes basées sur une partition de l'espace par des droites [1] par des voxels [2], ou par des plans [3], méthodes basées sur des maillages [4], etc.

A ce jour, la maquette du LURPA [5] basée sur le principe Zbuffer ou N-buffer [1] ne peut simuler la géométrie résultante que sur des portions très faibles de surface (quelques millimètres carrés pour une résolution au micromètre. Les défauts générés étant de différents ordres de grandeur, et la méthode de simulation fonctionnant à pas spatial et temporel fixes, il est impossible de combiner les différents effets pour obtenir à la fois une vue macroscopique des défauts et une vue ciblée. Aussi, les limitations des capacités de calculs et des méthodes de résolution utilisées actuellement restreignent le côté réaliste des simulations des écarts géométriques.



**Simulation CATIA** : zone de 100x100 mm, 30 min de calcul. Le maillage utilisé par la simulation est visible !!



**Simulation Nbuffer sur 1 cpu** : zone de 5x5 mm, 10 min de calcul, écarts géométriques simulés de l'ordre du micromètre.

Mais depuis peu, la société NVIDIA a développé CUDA, une architecture de traitement parallèle permettant de décupler les performances de calcul du système en exploitant la puissance des processeurs graphiques. Nous souhaitons donc exploiter cette architecture pour améliorer les temps de calcul et donc leur complexité. Cela nous permettrait par la suite d'enrichir les simulations de phénomènes réels plus complexes liés par exemple à l'utilisation de particules abrasives libres

Le défi est de pouvoir simuler les écarts géométriques générés par la succession de toutes les opérations d'usinage, avec des outils coupants et/ou des abrasifs, jusqu'à l'obtention d'états de surfaces polis miroir. Cette caractéristique de réflexion est atteinte lorsque les écarts géométriques sont de l'ordre de 50 nm, c'est à dire 10 fois inférieurs à la longueur d'onde de la lumière visible [5].

On peut donc définir 4 échelles de simulation tenant compte du type d'outil utilisé et de sa modélisation.

1. La première échelle est celle utilisée actuellement par les logiciels de FAO. L'outil est modélisé par une entité canonique (cylindre, sphère ou tore) et l'ordre des défauts est celui des paramètres de calcul des trajectoires, c'est-à-dire 10  $\mu\text{m}$  environ.
2. La seconde échelle consiste à modéliser l'outil et ses arêtes de coupe en intégrant la vitesse de déplacement et de rotation de l'outil pour simuler les traces laissées par les dents. Les détails sont ici simulés avec une précision de l'ordre de 1  $\mu\text{m}$ .  
À partir de cette échelle, les moyens de mesure des outils et des surfaces produites dont nous disposons ne sont pas suffisamment performants et des investissements sont nécessaires. Il sera alors possible de prendre en compte la géométrie réelle des outils et de comparer les résultats de simulation numérique avec les états de surfaces réels.
3. La troisième échelle est celle du grain abrasif de quelques dizaines de micromètres qui permet d'effectuer les opérations de prépolissage. Les particules sont agglomérées sur un support cylindrique, sphérique ou plan. Les détails sont de l'ordre de 100 nm.
4. Enfin, la dernière échelle est celle de l'abrasif libre d'un diamètre de quelques microns lors du polissage de finition. Il faut ici considérer une approche statistique pour évaluer le nombre de grains en contact avec la matière à chaque instant. Les détails simulés sont de l'ordre de 10 nm.

Pour cette échelle, nous ne disposons pas encore de méthode de simulation fiable, mais la réduction des temps de simulation nous aiderait à progresser plus rapidement.

Les techniques de contrôle des états de surfaces sont basées sur l'utilisation de capteurs sans contact qui mesurent l'altitude des points des surfaces. Ensuite, le nuage de points mesuré est post traité par des logiciels qui permettent le calcul des critères normalisés des états de surfaces. La simulation des topographies de surfaces que nous envisageons doit permettre de reconstituer ce nuage de points et s'appuyer ensuite sur les logiciels de post traitement (Mountain Map par exemple).

## Objectifs

Quelle que soit l'échelle, la technique de simulation que nous avons retenue est celle du N-Buffer [1]. Cela consiste à partitionner l'espace avec des droites normales à la surface considérée et à calculer les intersections de la trajectoire de l'outil avec ce réseau de droite. Dans la simulation, l'outil peut lui-même être représenté par un maillage dont la taille influe directement sur le temps de calcul, de même que la discrétisation de la surface et de la trajectoire. La méthode consiste à calculer les intersections entre le maillage de l'outil et le N-buffer pour des positions successives de l'outil représentant son mouvement le long de la trajectoire. L'opération de base est identique à celle effectuée lors des opérations de « ray-tracing », il s'agit du calcul de l'intersection d'un triangle et d'une droite. Le problème est donc massivement parallèle puisque dans un contexte purement cinématique, il n'existe aucune notion d'antériorité.

La difficulté réside dans :

- le découpage intelligent du problème pour tirer pleinement parti des millions de blocks de threads des GPU
- des latences lors des échanges de données entre mémoires vives CPU et GPU
- De l'exploitation optimale du peu de mémoire allouée par le GPU à chaque block de threads

En effet, pour exploiter au mieux le calcul sur GPU, et tirer parti pleinement de la puissance disponible sur ces processeurs, il faut adapter le traitement numérique des données. Sur les GPU, l'idée maîtresse est d'appliquer un même traitement sur des données différentes (acronyme SIMD : Single Instruction, Multiple Data).

Si l'on n'utilise pas cette possibilité, les performances s'effondrent puisque chaque processeur disponible sur une carte GPU (pris individuellement) est moins rapide que le CPU et dispose de moins de mémoire que le CPU. La seule solution pour accélérer les calculs est donc de maîtriser le traitement parallèle des données. A titre de comparaison, quand sur une carte CPU on dispose de quelques dizaines de cœurs, on en a facilement plusieurs centaines sur un GPU. Par exemple, le processeur Intel Xeon « Westmere » dispose de 12 cœurs, pendant qu'une carte graphique telle que la Nvidia Quadro 4000 dispose de 256 cœurs.

Ainsi l'exploitation des GPU demande une bonne expertise en programmation pour être proche du matériel, voire même de revisiter l'algorithme pour obtenir un gain important en temps de calcul. Il existe principalement deux environnements de développement sur GPU. Le premier proposé par Nvidia, nommé CUDA est très proche du langage C. Le second, OpenCL peut être utilisé pour tout type de processeur à plusieurs cœurs, mais nécessite un compilateur spécifique au matériel.

Enfin, on peut même imaginer exploiter des « clusters » de GPUs pour réduire encore les temps de calcul.

Le LMT Cachan dispose de plusieurs stations de travail équipées de cartes graphiques (plusieurs disposent de Quadro 4000 - 2 Go de RAM et 256 coeurs), et de plus sur le cluster 8 nœuds dit « graphiques » disposent chacun d'une carte GeForce GTX 260 - 896 Mo et 192 cœurs).

Le premier objectif est donc d'adapter l'algorithme à une carte GPU, en le couplant éventuellement avec une phase de visualisation. L'étape suivante étant de s'intéresser à l'extension à plusieurs GPU en faisant du parallélisme non seulement au sein des GPU mais aussi entre les GPU (utilisation des nœuds graphiques du cluster du

LMT Cachan). Nous nous orienterons naturellement vers l'environnement de développement CUDA.

De plus, nous souhaitons pouvoir affiner le résultat de simulation en temps réel lorsque l'utilisateur zoom sur la surface. Il s'agit de donc de faire une simulation multi échelles avec mise à jour dynamique du rendu. Il peut être alors nécessaire de stocker les informations déjà calculées à petite échelle pour augmenter la vitesse de rendu et d'utiliser des méthodes de simplifications pour passer aux plus grandes échelles comme les ondelettes par exemple [7].

Il est également possible d'utiliser les mêmes techniques de « ray-tracing » pour effectuer des lancers virtuels de rayons lumineux sur la surface simulée afin d'évaluer sa réflectivité. Ces méthodes sont utilisées pour effectuer des rendus réalistes dans les logiciels de CAO et de nombreux modèles existent dans la littérature [8]. Nous pourrions valider les simulations effectuées grâce à la mise à jour du moyen de contrôle sans contact dont dispose le LURPA. Il n'y a pas réellement de difficultés dans cette tâche, mais comme les techniques de calcul sont les mêmes, il serait dommage de ne pas aller jusque-là.

## Organisation

Ce projet, d'une durée de 2 ans, est porté par l'équipe suivante :

- Christophe Tournier, professeur, LURPA, responsable scientifique, 10 %.
- Pierre-Alain Boucard, professeur, LMT, responsable scientifique, 10 %.
- Sylvain Lavernhe, MCF, LURPA, 10%
- Hugo Leclerc, IR, LMT, 10 %
- Felix Abecassis, stagiaire de dernière année de l'EPITA, 6 mois, 100 %.

## Demands financières

Le montant total du projet est de 26 k€, dont 12 k€ de salaire pris sur fonds propres et l'aide demandée s'élève à 14 k€ :

- 9 k€ d'investissement pour l'achat d'un crayon optique de mesure,
- 2 k€ pour la mise à jour du logiciel de traitement des données topographiques Mountain,
- 3 k€ pour l'évolution des nœuds graphiques du cluster du LMT

## Références

- [1] R. Jerard, R. Drysdale, K. Hauck, B. Schaudt, and J. Magewick. Methods for detecting errors in numerically controlled machining of sculptured surfaces. *IEEE Computer Graphics and Applications*, 9(1):26–39, 1989.
- [2] D. Jang, K. Kim, and J. Jung. Voxel-based virtual multi-axis machining. *International Journal of Advanced Manufacturing Technology*, 16:709–713, 2000.
- [3] Y. Quinsat, L. Sabourin, and C. Lartigue. Surface topography in ball end milling process: Description of a 3d surface roughness parameter. *Journal of materials processing technology*, 195(1-3):135–143, 2008.
- [4] W. He and H. Bin. Simulation model for cnc machining of sculptured surface allowing different levels of detail. *The International Journal of Advanced Manufacturing Technology*, 33:1173–1179, 2007. 10.1007/s00170-006-0543-1.
- [5] Y. Quinsat, S. Lavernhe, and C. Lartigue. Characterization of 3d surface topography in 5-axis milling. *Wear*, 271(3-4):590–595, 2011.
- [6] P. Beckmann. Scattering by composite rough surfaces. *Proceedings of the IEEE*, 53(8):1012 – 1015, 1965.
- [7] X. Heurtebise and S. Thon. Multiresolution representation and deformation of very large volume datasets based on haar wavelets. In *Proceedings of the 2008 3rd International Conference on Geometric Modeling and Imaging*, pages 34–40. IEEE Computer Society, 2008.
- [8] K. E. Torrance, E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. In *Journal of Optical Society of America*, vol. 57, n°9, pp.1105-1114 (1967)