

CFA/VISHNO 2016

Méthode de développement participatif appliqué à un code de propagation acoustique : retour d'expérience du Hackathon Code_TYMPAN

P.-O. Chacun^a, B. Hamrouche^b et I. Schmich-Yamane^c

^aEDF DIPDE, 154, Avenue Thiers, CS60018, 69458 Lyon, France

^bEDF DTG, Service Opérationnel Thermique et Mécanique, 21 avenue de l'Europe, 38040 Grenoble Cedex 9, France

^cEDF - DTG, 21, avenue de l'Europe, 38000 Grenoble, France
pierre-olivier.chacun@edf.fr



LE MANS

Les méthodes de développement participatif, comme celle du hackathon, issues du monde du logiciel libre peinent à pénétrer dans les pratiques de conceptions traditionnelles, notamment pour les codes de propagation acoustique. Code TYMPAN, un code de propagation sonore en milieu extérieur développé par EDF, est un logiciel téléchargeable librement qui permet à chacun de développer les modules complémentaires adaptés à son utilisation. Cet article présente un retour d'expérience sur le hackathon organisé en octobre 2015 par l'équipe EDF du logiciel Code TYMPAN à Paris. L'événement a été organisé en deux phases : une rencontre de préparation qui a permis de présenter les avancées du code et le principe du hackathon ainsi que de définir quels projets seraient développés, suivie de deux jours de travail collaboratifs sous forme de « task force ». La contrainte de délai de réalisation extrêmement court permet de focaliser l'énergie sur le résultat final, exclut ce qui pourrait retarder la réalisation (étude de faisabilité, note de principe, etc.) et maintient l'attention sur les tâches essentielles du projet. Cette organisation a permis d'atteindre une efficacité dans la production d'idée et de maquettage de développement qu'il semble difficile d'atteindre dans des relations de travail plus conventionnelles. Les sujets développés concernaient la modélisation de sources mobiles comme par exemple le déplacement d'un avion sur une piste d'aéroport, le lancement de calculs multiples afin de pouvoir effectuer des études paramétriques et le prototypage d'un module de propagation d'acoustique intérieure adapté aux locaux industriels. L'objectif pour EDF était de profiter de l'événement pour dynamiser la communauté Open Source et de profiter des développements réalisés, mais le bénéfice est bien plus large puisque d'autres industriels pourront tirer profit de ces travaux.

1 Introduction

Code_TYMPAN est un logiciel de propagation acoustique en milieu extérieur développé par EDF pour évaluer l'impact acoustique de ses sites industriels. Après dix années de développements et d'utilisation exclusivement internes, il a été décidé de diffuser le logiciel en libre. Une telle décision implique généralement une attente particulière de la part du créateur du logiciel : celle d'avoir une communauté qui contribue à la vie de celui-ci, soit par des retours d'utilisation (idées, relevés de bugs, critiques des méthodes...) ou par des développements qui permettront d'améliorer les fonctionnalités existantes ou de les faire évoluer.

Les développements réalisés ces dernières années sur Code_TYMPAN s'avèrent particulièrement propices à ce type de fonctionnement. En effet, ceux-ci ont permis de le transformer architecturalement d'un bloc monolithique difficile à comprendre et à prendre en main, en un outil composé de blocs découplés que les utilisateurs ou développeurs peuvent soit explorer, soit considérer comme des boîtes noires suivant leurs besoins pour étendre les fonctionnalités du logiciel.

Afin de dynamiser la communauté open source, tout en tirant profit de la nouvelle architecture en réalisation de prototypes, l'équipe acoustique d'EDF a organisé fin 2015 un hackathon. Réunion intense où les participants écrivent du code, le hackathon propose de nouvelles manières d'aborder les problématiques de recherche et développement informatique en se focalisant sur le résultat final, c'est-à-dire sur un démonstrateur fonctionnel répondant aux besoins exprimés par le client, le commanditaire, ou les utilisateurs de l'équipe.

L'objectif de cet article est de faire un retour d'expérience sur l'organisation et les résultats d'un hackathon, en termes de travail collaboratif et en termes de prototypes ressortis, pour travailler sur un logiciel d'acoustique comme Code_TYMPAN.

2 Présentation de Code_TYMPAN

Code_TYMPAN est un logiciel de propagation acoustique dans des scènes 3D complexes destiné à l'évaluation et la prévision de l'impact sonore de sites industriels. Il a été développé par EDF R&D au début des années 2000 pour permettre la réalisation d'études d'ingénierie en acoustique environnementale.

2.1 Présentation du logiciel

L'entrée en vigueur d'une réglementation des installations nucléaires en 1999 a nécessité pour EDF de disposer d'un outil moderne d'acoustique environnementale pour évaluer la conformité de ses sites industriels et dimensionner les solutions correctives éventuellement nécessaires. Les produits proposés sur le marché étant essentiellement orientés vers le traitement des bruits de transport et mal adaptés au traitement des bruits industriels, EDF a décidé de concevoir une application dédiée : Code_TYMPAN.

Code_TYMPAN est donc un outil principalement dédié à la problématique de la propagation du bruit industriel dans l'environnement. Il possède une interface homme-machine riche permettant la construction 3D de modèles réalistes et un modèle métier (ou structure de données) conçu pour assurer la durabilité des modèles géométriques afin qu'ils puissent être stockés, réutilisés et étendus (une contrainte inhérente à la problématique du bruit industriel qui nécessite des études sur le long terme). Les solveurs de calcul intégrés sont aptes à répondre aux problèmes acoustiques posés par les installations industrielles de grandes dimensions en milieu rural. En effet, ceux-ci permettent de prendre en compte efficacement l'influence du milieu sur la propagation, en particulier les conditions météorologiques (gradients de vent et de température) et leur variabilité statistique, ainsi que la géométrie et les matériaux environnant le site (reliefs, sols).

2.2 Description des méthodes

Code_TYMPAN donne accès à différentes méthodes de simulation à travers deux solveurs acoustique [1]. Ces simulations peuvent être décomposées en une étape géométrique et une étape acoustique. L'étape géométrique consiste en la recherche de chemins acoustiques entre chaque couple source-récepteur. L'étape acoustique permet de calculer l'atténuation sur chaque chemin identifié dans la phase géométrique.

Actuellement, Code_TYMPAN dispose de deux solveurs acoustiques : Default Solver et Anime3D Solver.

Default Solver est le solveur historique, il a la particularité d'être utilisé pour les cas les plus simples et

pour les études les plus élémentaires. Il est basé, pour la partie géométrique sur une méthode de recherche d'enveloppes convexes [2] et pour la partie acoustique sur l'ISO9613 et la NMPB. (Voir figures 1-a et 2-a).

Anime3D Solver (pour Assessing Noise on the ENvironment in 3D) est un solveur plus avancé qui a pour finalité le traitement de situations plus complexes et inclus des traitements météo et la prise en compte de réflexions multiples et diffractions. Il se base pour la partie géométrique sur un lancer de rayons 3D optimisé [3] et pour la partie acoustique sur un calcul acoustique qui prend en compte les effets météo et l'effet de sols inhomogènes [4]. (Voir figures 1-b et 2-b).

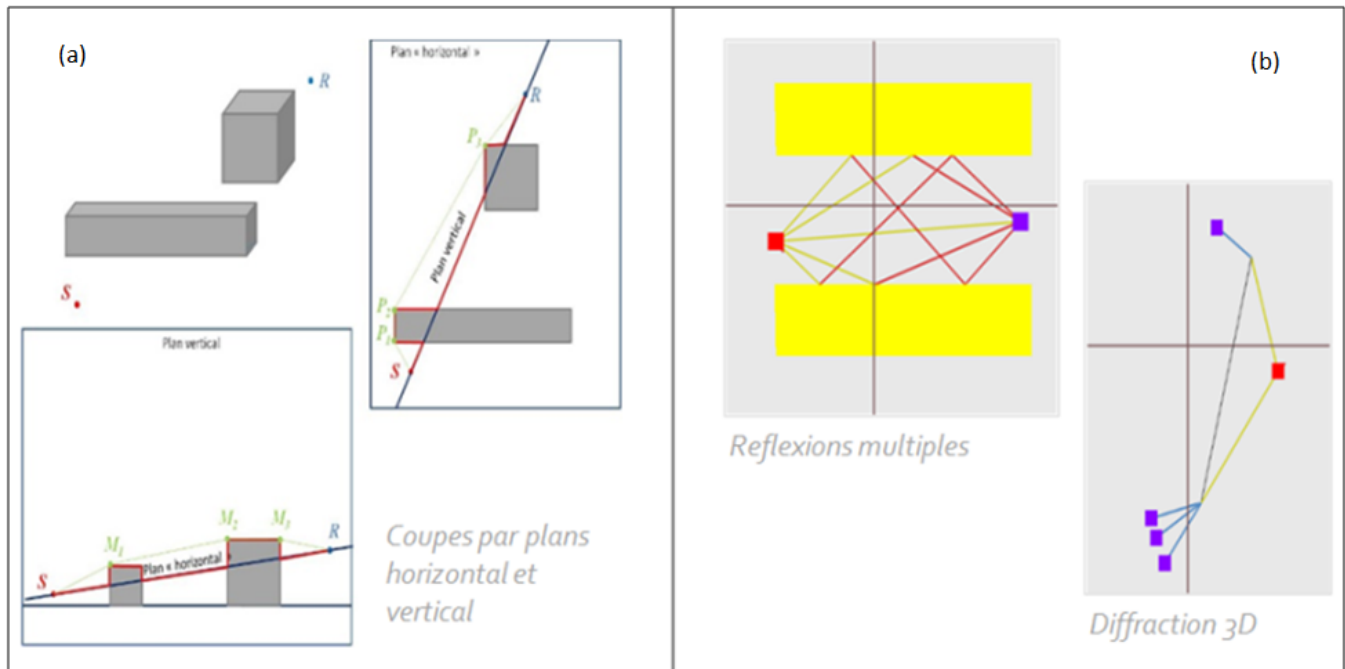


Figure 1 : Présentation de l'étape géométrique d'un solveur de Code_TYMPAN (2.a étape géométrique du Default Solver basée sur une recherche d'enveloppes convexes. 2.b étape géométrique de Anime3D Solver basée un lancer de rayons 3D optimisé)

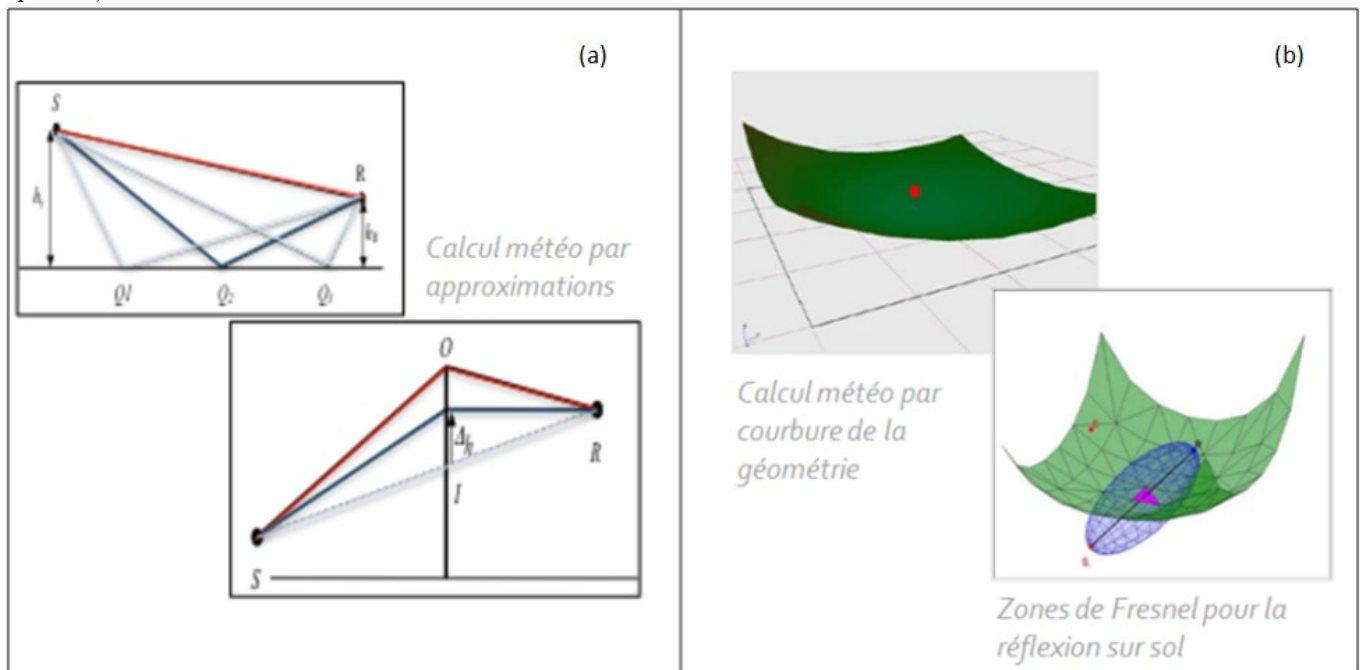


Figure 2 : Présentation de l'étape acoustique d'un solveur de Code_TYMPAN (2.a étape acoustique du Default Solver basée sur l'ISO 9613 2.b étape acoustique d'Anime3D Solver qui étend le calcul Défaut Solver avec un calcul météo plus précis et la prise en compte de sols hétérogènes)

2.3 Diffusion Open Source

Code TYMPAN est diffusé en Open Source licence GNU GPL v3 depuis juin 2012. Sa diffusion en tant que logiciel libre est un évènement important dans le développement de l'outil, puisqu'elle signe une volonté d'utilisation non plus seulement au niveau d'EDF, mais également auprès d'autres industries, de bureaux d'études et de laboratoires de recherche avec l'ambition d'être également un outil utilisable pour l'enseignement.

La diffusion libre permet également pour l'équipe acoustique d'obtenir un certain nombre de retours :

- Une utilisations plus large en nombre mais aussi en nature, des usages plus nombreux et variés (éventuellement différents des usages EDF) ;
- Des benchmarks (comparaisons de performances, de fonctionnalités, etc.) réalisés par les nouveaux utilisateurs avec leurs codes comparables à Code_TYMPAN sur leurs propres études ;
- Des enrichissements de l'application par des contributions de tiers et des projets de co-développement.

Pour atteindre ces objectifs, il est nécessaire d'avoir un code qui se prête bien à un travail collaboratif. Or dans son architecture originale, Code_TYMPAN offrait peu de flexibilité pour des développements externes.

2.4 Nouvelle architecture du logiciel

Code_TYMPAN a été historiquement développé en C++ comme une application monolithique adaptée aux besoins de l'ingénierie d'EDF. Du point de vue de l'architecture du code, les solveurs étaient intégrés dans une interface métier qui était le seul point d'entrée de l'utilisateur à travers son interaction avec l'interface graphique. Il résultait de cette architecture une extrême lourdeur dans la maintenance du code et tout particulièrement de son interface graphique et rendait une prise en main par la communauté open source très difficile puisqu'il fallait comprendre le code dans sa globalité (interface graphique, modèle métier et solveurs) pour espérer faire des contributions en C++.

Les travaux effectués ces deux dernières années par EDF ont permis de faire notablement évoluer Code_TYMPAN en le faisant passer de l'état d'un logiciel fermé et peu évolutif à une véritable plateforme de développement de méthodes de calcul acoustiques.

La version 4 de Code_TYMPAN, actuellement disponible à la communauté open source possède ainsi une architecture complètement différente. Les parties « solveur » et « métier » ont été séparées et transformées en briques élémentaires qui communiquent via une interface de programmation (API pour Application Programming Interface) en langage Python (voir figure 3). Il est donc désormais possible pour un utilisateur avancé de développer ses propres outils grâce à des scripts Python.

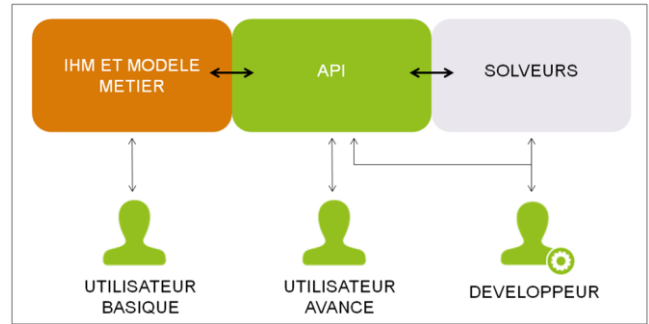


Figure 3 : Schéma de principe de la version 4 de Code_TYMPAN : des briques élémentaires qui communiquent via une API Python

L'existence de cette API Python permet par exemple de générer automatiquement des modèles 3D (création automatisée d'objets acoustiques, de points de calcul, importation de topographie, etc.), d'appeler le solveur sans ouvrir l'application pour lancer des calculs en série ou de récupérer certains objets pour réaliser des post-traitements, voire construire son propre solveur acoustique. L'utilisation de Python permet de profiter de nombreuses bibliothèques autour de ce langage qui sont disponibles gratuitement.

Ces fonctionnalités rendent le logiciel particulièrement adapté à une utilisation open source et au travail collaboratif.

3 Le hackathon

Dans cette optique, l'équipe a décidé de s'orienter vers des méthodes de travail moins conventionnelles : une présentation simple ne démontrerait pas l'utilité de la nouvelle API et des réunions classiques prendraient beaucoup trop de temps. L'idée est alors venue d'organiser un hackathon afin d'évaluer toutes les possibilités, accessibles facilement avec Code_TYMPAN et d'en profiter pour stimuler la communauté open source pour lui présenter les avantages de la nouvelle architecture.

3.1 Principe et objectifs

Un hackathon est un évènement où des développeurs se réunissent pour faire de la programmation informatique collaborative, sur plusieurs jours. Le terme, popularisé par la communauté d'openBSD et aujourd'hui très répandu chez la communauté open source [5], est un mot-valise constitué de « hacking » et « marathon ».

C'est à l'origine un rassemblement de développeurs organisés par équipe autour de porteurs de projet avec l'objectif de produire un prototype d'application prêt à fonctionner en quelques heures. Généralement sous forme de concours chronométré, l'équipe gagnante est désignée par un jury à l'issue du temps imparti. La référence au Marathon se justifie par le travail sans interruption des développeurs pendant plusieurs jours.

Souvent utilisé pour des développements purement informatiques, notamment pour des applications mobiles, cette méthode est utilisée de plus en plus pour réaliser des développements d'ingénierie pointus, nécessitant de recourir à des experts du domaine [6]. Cela permet de s'affranchir des lourdeurs inhérentes aux pratiques habituelles (étude d'opportunité, cahier de spécification technique, études de faisabilités, etc.), pour se concentrer

uniquement sur l'innovation et la démonstration de faisabilité (ou POC pour Proof Of Concept). Le délai de réalisation extrêmement court introduit un stress positif dans les équipes qui focalise les participants sur le résultat final et exclu de facto tout ce qui pourrait retarder la réalisation. L'environnement relaxé et enclin à tolérer l'échec est quant à lui propice pour faire émerger l'innovation [5].

En plus de l'objectif évident centré sur les résultats, le hackathon est surtout particulièrement adapté pour le développement de communautés, car il encourage les interactions et produit un environnement adapté à l'utilisation de méthodes agiles. D'ailleurs, les motivations premières des participants sont plus souvent l'apprentissage et le réseau plutôt que le prototypage en lui-même [5, 6].

3.2 Organisation

L'évènement a démarré par la tenue, le 22 septembre 2015, d'une réunion informelle de 2 heures (*meet-up*) en début de soirée pour présenter la version 4.0 de Code_TYMPAN, ses possibilités et le principe du hackathon. Des thématiques très larges ont été proposées comme exemples, mais les discussions ont d'elles-mêmes conduit à définir celles répondant aux besoins des participants : le bruit éolien, les sources mobiles, la modélisation d'objets complexes (insertion de routes, d'écran spécifiques), l'acoustique intérieure, le recalage et le calcul inverse.

Quatre sujets ont ensuite été retenus par EDF, selon leur popularité parmi les participants et les capacités de l'API à les traiter dans des délais raisonnables : bruit éolien, sources mobiles, acoustique intérieure et optimisation / recalage / étude de sensibilité.

Le Hackathon s'est déroulé les 5 et 6 octobre 2015. Les équipes ont été formées suivant les affinités de chacun. Elles ont travaillé sur une des thématiques retenues pour cibler un sujet de développement précis de pour réaliser un prototype de solution basé sur Code_TYMPAN. L'aspect compétition n'a pas été conservé, pour permettre l'échange entre les équipes.

Les trois équipes étaient composées d'acousticiens, de développeurs et d'utilisateurs actuels ou futurs de Code_TYMPAN.

Le déroulement des deux jours était ensuite rythmé par des phases successives de travail de groupe et points collectifs (*stand-up*) toutes les 3 heures pendant lesquels les trois équipes faisaient part de leur avancement, s'astreignaient à présenter des résultats opérationnels et pouvaient échanger sur leurs difficultés et expériences.

4 Projets développés

Les trois projets (outils de calculs multiples, sources mobiles, acoustique intérieure) qui ont été développés sont présentés dans les paragraphes suivants.

4.1 Outil de calculs multiples

L'objectif était de développer un utilitaire permettant de lancer des calculs en série pour plusieurs applications :

- le lancement de calculs multiples à partir d'un ensemble de valeurs de paramètres : calcul suivant différents régime de vent dans l'éolien, scénarii d'insonorisation, etc. ;
- des études de sensibilité d'un modèle autour d'un jeu de paramètres (niveau de puissance acoustique des sources, hauteur des buttes...), ce qui permet d'estimer la robustesse des recalages et les paramètres les plus influents du modèle.
- la propagation d'incertitude par méthode Monte-Carlo [7], ce qui permet d'estimer la fiabilité des prédictions d'un calcul acoustique (figure 4).

En terme d'API, ce travail touche à tous les blocs de l'architecture logicielle et permet de jouer à la fois sur les paramètres du modèle métier (position des sources, hauteurs des bâtiments, ...) et les paramètres du calcul solveur (gradient de vent, nombre de rayons lancés, ...).

A la fin du deuxième jour, l'équipe disposait d'un script permettant de créer les plans d'expérience, d'effectuer les calculs et de visualiser les résultats pour les trois types d'applications.

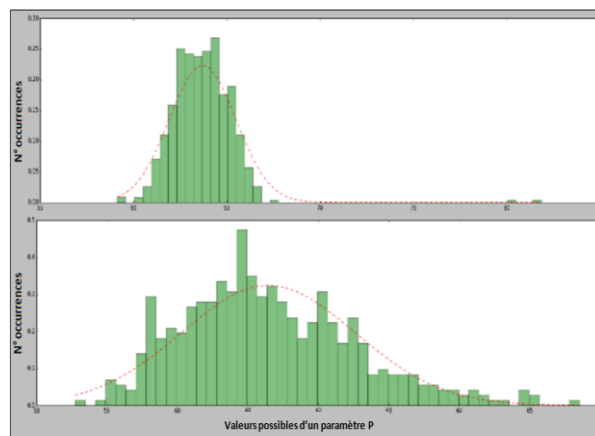


Figure 4 : Histogrammes pour visualiser les résultats de la propagation d'erreurs par Monte-Carlo.

Bien qu'étant dans la continuité de l'utilisation classique de Code_TYMPAN, ce travail est une première étape vers plusieurs applications intéressantes dans l'évaluation, le recalage et la propagation d'erreurs dans des modèles et peut facilement intéresser d'autres communautés d'utilisateurs.

Ce développement a donné naissance à deux partenariats concrets, tous les membres de l'équipe souhaitant poursuivre dans la mise en place d'un tel outil. Ceci n'est pas surprenant, vu l'importance aujourd'hui de l'estimation des incertitudes et du besoin de modèles de plus en plus précis.

4.2 Sources mobiles

L'objectif de ce projet était de pouvoir déplacer un ensemble de sources de bruit selon une trajectoire définie et de calculer la contribution sonore totale en un ou plusieurs récepteurs.

Du point de vue de l'API, ce travail se focalise sur la modification du modèle métier pour automatiser les changements de positions et la caractérisation des sources.

Il utilise le solveur comme une boîte noire uniquement pour lancer des calculs.

A la fin du deuxième jour, l'équipe disposait d'un script permettant de déplacer des sources selon une trajectoire et effectuer des rotations, de calculer la contribution aux points récepteurs pour chaque position de source et de récupérer une liste de toutes les contributions en dehors de l'IHM en moins d'une minute, ce qui a l'avantage d'être rapide, automatique et complètement paramétrable, moyennant une modification du script Python.

Suite à quelques difficultés logicielles, les résultats obtenus n'ont pu être tracés. Toutefois, l'interface Python permettra de visualiser les résultats sous différents types de graphes tels que le niveau de pression sonore en fonction des positions des sources.

Le déplacement des sources sort du cadre d'utilisation classique de Code_TYMPAN pour répondre à des besoins du domaine du bruit des infrastructures de transport en simulant la contribution liée au déplacement de véhicules (véhicules légers, poids lourds, avions, trains) ou d'engins de chantier. Il trouve également des applications dans le domaine éolien, avec la mise en rotation de différentes sources complexes (pales notamment). De par sa structure, le déplacement d'une source dans Code_TYMPAN peut facilement être transposé aux récepteurs. Cela ouvre des possibilités de calcul sur l'exposition des personnes en mouvement (problématique du bruit au travail).

4.3 Acoustique intérieure

L'objectif de cette équipe était d'utiliser le lancer de rayons présent dans Anime3D Solver pour réaliser un module de calcul d'acoustique intérieure dans des locaux de forme complexe, en intégrant à la fois les effets de réflexion et de diffraction.

En terme d'API, ce développement démontre la flexibilité de la nouvelle architecture de Code_TYMPAN pour se passer du modèle métier disponible, créer une géométrie dans un nouvel outil (google SketchUp en l'occurrence) et de ne se servir que de la partie géométrique d'Anime3D Solver.

A partir d'une position de source, la propagation géométrique est réalisée avec le lancer de rayon optimisé avec un ordre de réflexion et de diffraction choisis. La réflexion sur les surfaces se fait de façon spéculaire jusqu'à l'arrivée dans la sphère réceptrice. Aux récepteurs ponctuels ou pour des cartographies les différents trajets sont sommés de façon énergétique sans conservation du temps de trajet. Dans un premier temps les matériaux ont été supposés tous totalement réfléchissants. L'absorption acoustique des matériaux devra être prise en compte dans un deuxième temps.

A la fin du deuxième jour, l'équipe disposait :

- De la modélisation d'une géométrie complexe sous Google SketchUp, de sa correction sous MeshLab, de l'export de la géométrie en format PLY et de l'insertion dans le fichier PLY des coordonnées des sources et des récepteurs ;

- D'un script Python utilisant l'API Code_TYMPAN pour piloter le lancer de rayons optimisé (prise en compte des réflexions multiples et de la diffraction) et d'accéder à l'ensemble des paramètres de ce module. Ce script permettait de réaliser un calcul acoustique à partir des rayons reçus en chaque récepteur (voir figure 5), de calculer une cartographie horizontale (voir figure 6).

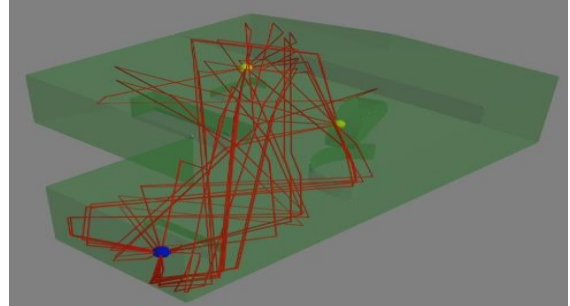


Figure 5 : Exemple de visualisation des rayons issus de 2 sources (en jaune) et parvenant à un récepteur, en passant par le plafond uniquement avec des réflexions jusqu'à l'ordre 6 dans un volume complexe contenant des éléments courbes.

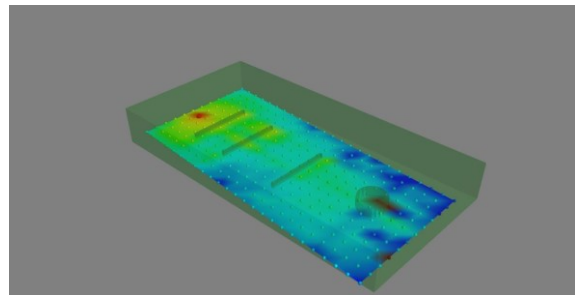


Figure 6 : Exemple de visualisation d'une cartographie de niveaux sonores, pour une source dans un volume fermé avec maximum trois diffractions.

Ce développement sort complètement du cadre d'utilisation classique de Code_TYMPAN et permet de récupérer des éléments développés pour de l'acoustique environnementale et de les étendre à l'acoustique intérieure.

5 Bilan et retour d'expérience

L'organisation d'un hackathon a permis d'atteindre une efficacité remarquable dans la production d'idées et de maquettage de développement d'outils d'ingénierie acoustique pour répondre aux besoins des utilisateurs présents dans chaque équipe.

Tout d'abord la mise en place d'une phase de meet-up a permis d'optimiser la sollicitation des partenaires potentiels et des équipes (2h en début de soirée) et a laissé une grande place aux échanges grâce à une présentation démo courte (45'), incluant la description et l'explication des objectifs du hackathon. La production d'idées et l'expression des besoins autour du développement et de l'utilisation de la plateforme Code_TYMPAN ont été productifs, malgré la présence de plus de vingt personnes.

Le hackathon en lui-même a donné lieu au développement de prototypes répondant à des besoins concrets du métier exprimés par un panel large et qui donneront lieu à industrialisation si le besoin s'en fait sentir. Il a permis de rassembler des équipes pluridisciplinaires autour de problématiques communes. Le développement de plusieurs projets de front par des équipes différentes, forcées à échanger à intervalles réguliers, a également suscité des synergies entre équipes et des échanges sur les développements réalisés.

Au-delà des résultats immédiats, le hackathon a permis de faire collaborer différents acteurs sur des thématiques qui présentent un intérêt commun : de ces collaborations, plusieurs partenariats sont nés. Point encore plus intéressant, de par le large panel des participants, les sujets abordés ont naturellement fait ressortir les tendances et besoins communs présents aujourd'hui dans le domaine de l'acoustique (notamment en termes de calcul paramétrique et de recalage de modèles). Ils ont également permis aux différents acteurs de différents domaines de recouper leur besoins et d'exploiter des outils développés dans un cadre différent (celui de l'acoustique environnementale) pour les résoudre.

Les résultats sont convainquants pour tous les participants et le hackathon est retenu comme un très bon format de travail collaboratif pour la création de partenariat et le prototypage d'outils. Cela dit, pour permettre une bonne rentabilité de l'événement, une bonne préparation est de mise pour assurer que l'événement soit convivial et productif.

5 Conclusion

Le hackathon s'avère donc être un outil particulièrement puissant pour faire du travail collaboratif et initier des partenariats et des développements de prototypes.

Il semble aussi que la communauté acoustique, qui est en connexion avec beaucoup d'autres domaines variés puisse y trouver un moyen très adapté pour stimuler la collaboration entre domaines et tirer le maximum des compétences de chacun dans le but d'imaginer de nouvelles réalisations utiles à tous.

Ceci est particulièrement vrai lorsque l'événement se fait autour d'une application open source offrant la possibilité de ne pas brider l'imagination des participants et d'encourager l'innovation. Pour cela, la nouvelle architecture de Code_TYMPAN s'est révélée très utile, permettant facilement d'améliorer des fonctionnalités existantes suivant le besoin ou d'en récupérer pour étendre l'utilisation du logiciel.

Les futurs développements sur Code TYMPAN essayeront de poursuivre ce travail de « généralisation » des fonctionnalités à la fois pour ouvrir des portes à des applications futures et pour continuer à stimuler le travail collaboratif avec une communauté open source diversifiée.

Remerciements

Pour cet événement, EDF était assistée par la société af83 qui s'est chargée de l'organisation logistique de l'événement et de la mobilisation des participants.

Pour les développements sur Code_TYMPAN et notamment sur la mise en place de l'API Python, EDF s'est appuyée sur la société Logilab.

Références

- [1] S. Le Bourdier et D. Thomasson, *Code_TYMPAN open source software dedicated to the calculation of industrial noise in the environment*, *Internoise, Innsbruck (2013)*
- [2] Description des modèles de DefaultSolver [online] www.code-tympan.org
- [3] F. Junker, *ANIME3D: A full 3D method for calculating the impact of industrial noise on the environment*, *Internoise, Innsbruck (2013)*
- [4] M. Dreher, G. Dutilleul, and F. Junker, *Optimized 3D ray tracing algorithm for environmental acoustic studies*, in *Acoustics 2012, 2012*.
- [5] H.Lapp and co: *The 2006 NESCent Phyloinformatics Hackathon: A Field Report*
- [6] G. Briscoe, C. Mulligan, *Digital Innovation: The Hackathon Phenomenon, 2014*
- [7] Nicholas Metropolis et Stanislaw Ulam, *The Monte Carlo Method*, *Journal of the American Statistical Association*, vol. 44, no 247, septembre 1949, p. 335-341