



Une interface graphique, ouverte et évolutive, pour les codes de calcul en propagation acoustique

J. Picaut (LCPC)
N. Fortin (Développeur, LCPC-ULR)



Ce travail a bénéficié du soutien financié de l'ADEME (Projet OPALHA)



Contexte



- Projet OPALHA (2007-2011): développement d'un outil de prévision acoustique pour l'habitat et le milieu urbain (modèle de diffusion)
 - Université de La Rochelle (pilote) : A. Sakout, A. Billon*, N. Fortinµ
 - Laboratoire Central des Ponts et Chaussées : J. Picaut
 - Université de Poitiers : V. Valeau
 - SerdB (BET Nantes) : T. Legouis
 - ADEME : E. Thibier, P. André

* Chercheur associé, Université de Liège ^µ Personnel ULR en poste au LCPC



Contexte



Objectifs du projet :

 Développement d'un code de calcul « acoustique » basé sur la résolution d'équation de diffusion



Développement d'une interface



 Utilisation du logiciel COMSOL Multiphysics® pour la résolution d'équations de transport



Contexte

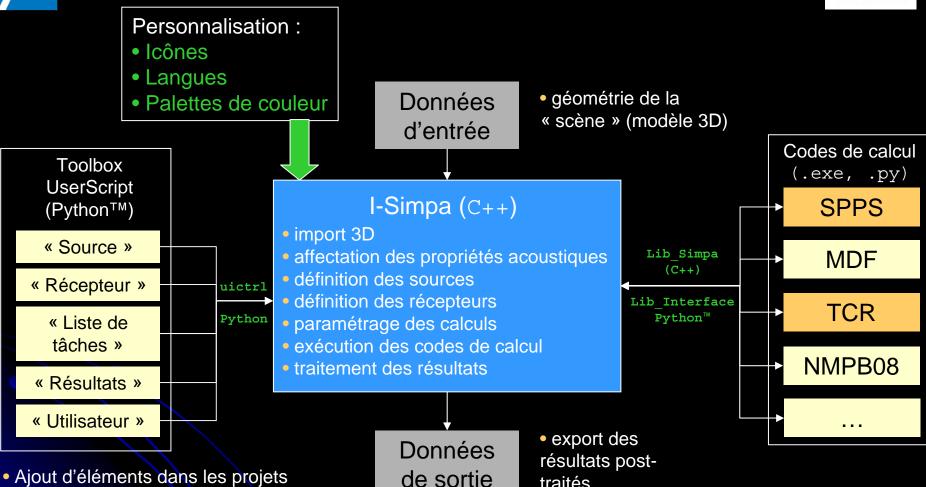


• Généralisation de l'interface :

- Commune pour tout code de propagation
 - Facilite la comparaison/validation/développement
 - Simplifie la gestion des données d'entrée
 - Simplifie le traitement des résultats
- « Ouverte » et « évolutive »
- Documentée
- Diffusable (création d'une communauté)







- Ajout d'éléments dans les projets
- Ajout de fonctions dans les menus
- Manipulation des résultats
- Manipulation d'éléments...

traités





Interface « ouverte » :

- Import des géométries au format 3DS (Google Sketchup Pro)
- Import BDD de matériaux acoustiques
- Représentation des résultats et paramètres en feuilles de calcul
- Copier/Coller de toutes les données vers logiciels tiers (tableur)
- Reconnaissance des formats de fichier et exécution associée
- Formats de fichier XML (calcul) et GABE (LCPC) documentés
- Archivage des résultats/calculs avec copie des fichiers de conf.
- Console Python™ intégrée
- Documentation importante :
 - Documentation « développeur »
 - Documentation des librairies
 - Manuel d'utilisation de l'interface (en cours)





Interface « évolutive » :

- Ajout de codes de calcul « tiers » (Lib_Simpa, Lib_interface)
- Ajout d'éléments (éléments, propriétés…) avec des scripts Python™ (librairie uictrl)
- Ajout de fonctions (menu, traitement des paramètres, traitement des résultats) avec des scripts Python™ (librairie uictrl)
- Interface « manipulable »
- Interface multilingue (fichiers de langue)
- Graphisme personnalisable (icônes, palettes de couleur)

