

Arithmétique flottante & Codes de calcul

Retours d'expériences

ANEEO : UN CABINET DE CONSEIL ET DE TECHNOLOGIES

Dont la valeur est apportée par les hommes et les femmes...

200 personnalités
aux expertises
variées



Une organisation
aplatie favorisant
les **initiatives**



1 projet ANR
1 projet FUI
1 thèse en cours
7 docteurs

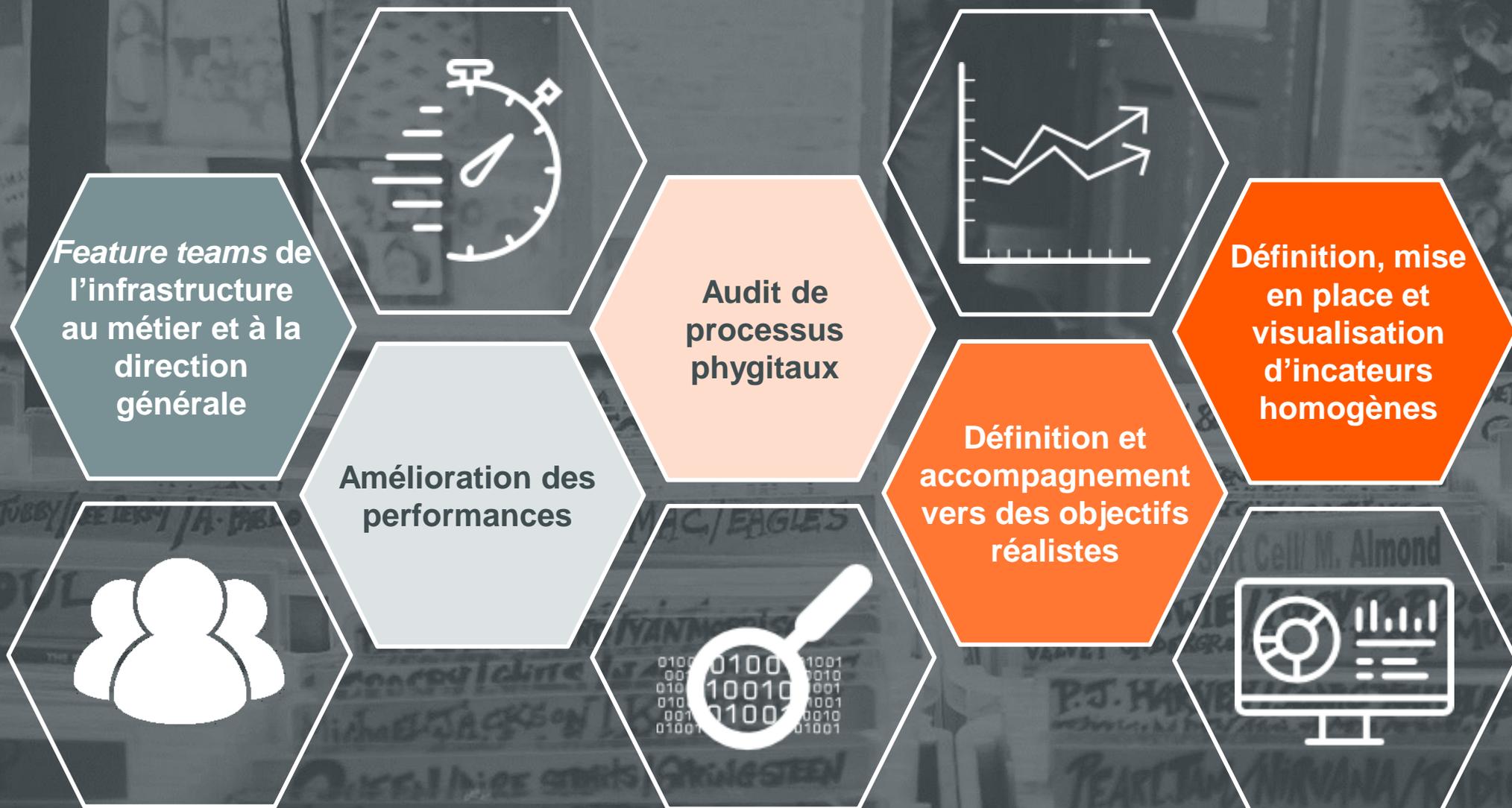


22 M€ de
Chiffre d'affaire

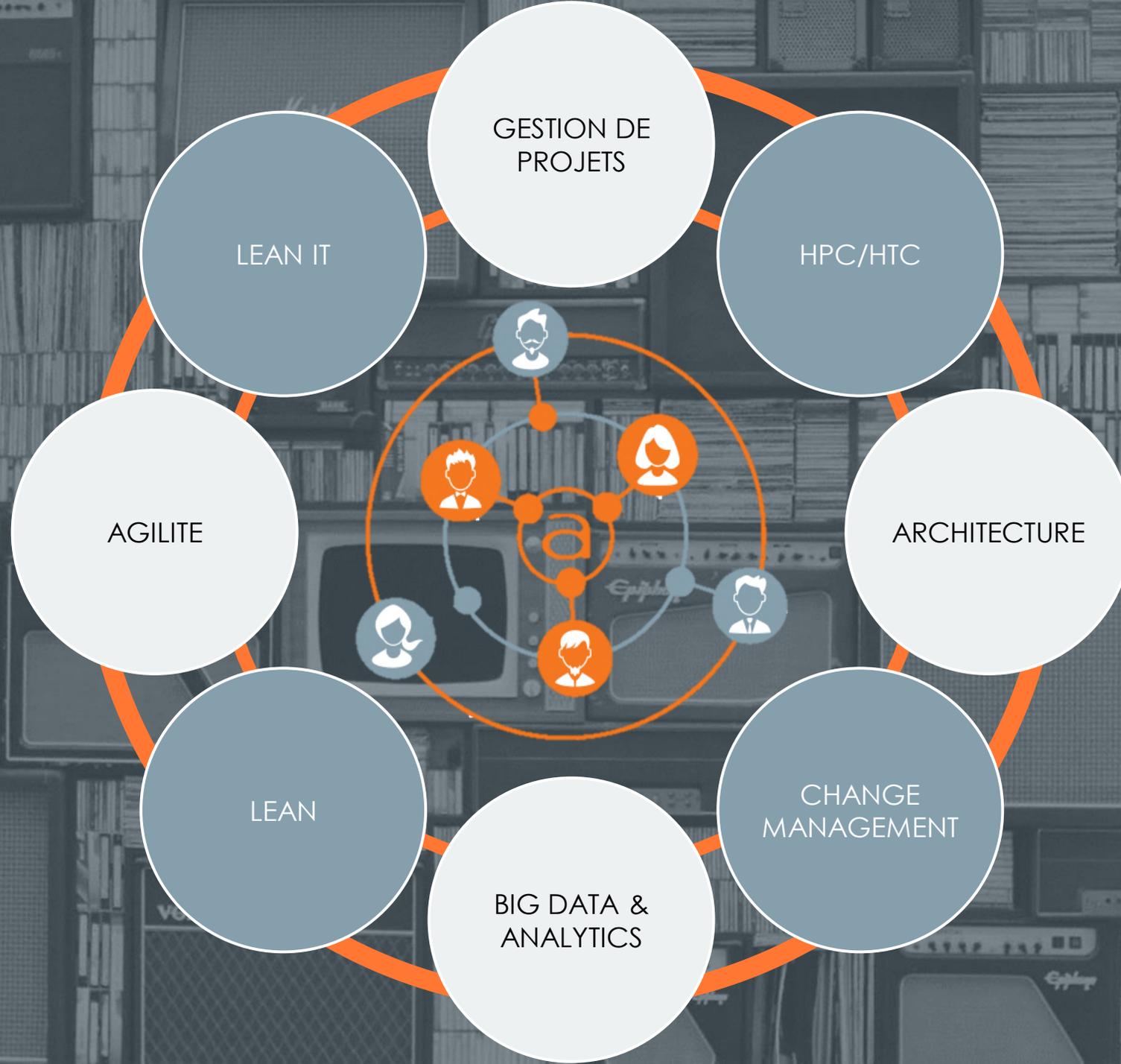


... pour répondre à vos enjeux opérationnels

QUELQUES **ACTIVITÉS** D'ANEO

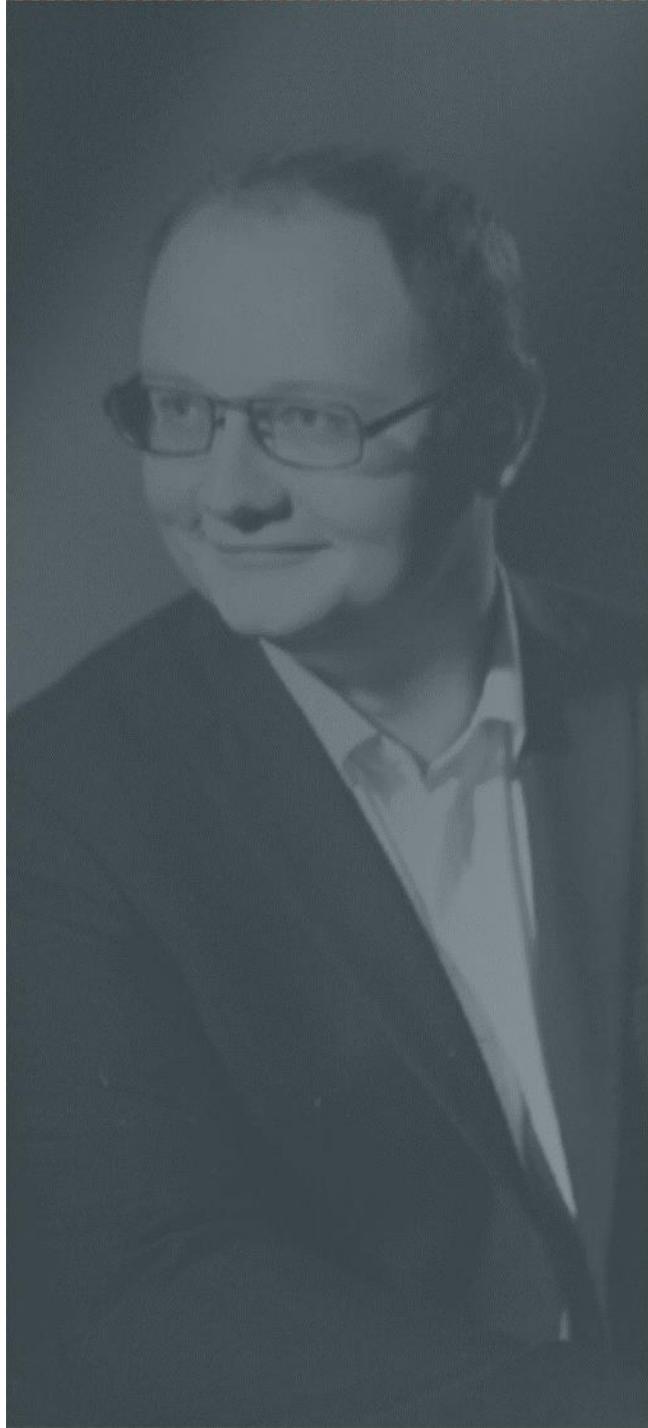


ORGANISATION ANEO



**WILFRIED
KIRSCHENMANN**
MANAGER@ANEO

5



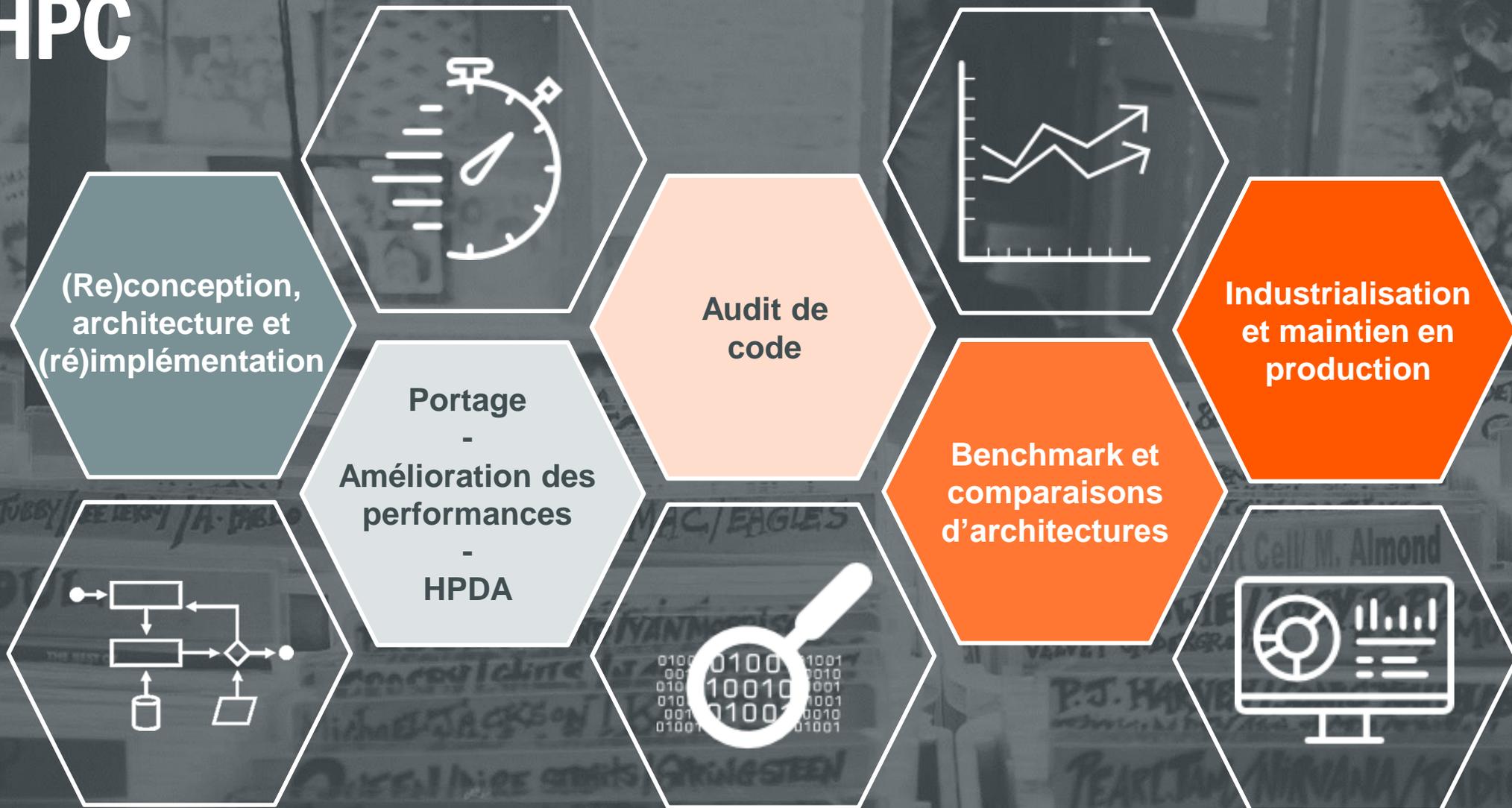
HPC/HTC
Animateur du cercle
Expert

Big Data & Analytics
Animateur du cercle

Intensive Cloud Computing
Pilote des activités
Expert

Recherche & Innovation
Pilote des activités (1 projet ANR, 1 projet FUI, R&I client)
Encadrant d'une thèse en science des organisations

NOS **MISSIONS** TYPE DANS LE HPC



Audit qualité et maintenabilité d'une bibliothèque

7



Code de valorisation de portefeuille

Utilisé pour définir les stratégies de placement d'assurances-vie
et répondre aux exigences réglementaires

Code distribué sur grille depuis une feuille
Excel

Il s'agit au départ du portage en C d'un code VBA

Code distribué sur grille

Distribution des tâches indépendantes exécutées à travers des
instances locales d'Excel

55k lignes de code C

68 fichiers

Environ 25% de commentaires

Une bibliothèque de calcul sous Excel

Première difficulté : exécuter le code sans erreurs

Afin de pouvoir nous assurer des résultats que l'on propose, il est important de pouvoir exécuter le code dans notre environnement

Quelques difficultés rencontrées annoncent la couleur

- 1 Les résultats changent selon que l'on utilise Excel x86 ou x64
- 2 La même DLL ne fournit pas les mêmes résultats exécutée depuis Excel ou via un exécutable console
- 3 Lorsque l'on compile pour l'unité flottante à SSE2, les résultats sont identiques sous Excel et via un exécutable

Excel modifie le mode d'arrondi par défaut de l'unité X87

Excel définit le mode d'arrondi de l'unité X87 vers $+\infty$ mais ne touche pas à l'unité SSE

Si la bibliothèque utilise l'unité x87, son comportement ne sera donc pas le même dans Excel et en dehors d'Excel

La version X64 d'Excel ne modifie pas le mode d'arrondi

En x64, par défaut, c'est l'unité SSE2 qui est utilisée

Des dépassements de tableaux mais dans des zones allouées

Certains accès en écriture/lecture sont effectués avec des indices plus grands que la dimension courante du tableau

Pas de résultat aléatoire car les variables sont initialisées

Les optimisations du compilateurs conduisent à des comportement différents car ils peuvent conduire à supprimer des écritures censées ne servir à rien... Sauf pour les effets de bord sur les autres tableaux

On peut retrouver ces éléments en initialisant tout les éléments avec des NaN et en interceptant les exceptions

Changer de compilateur

Un compilateur bien configuré peut améliorer les performances sans nuire à la qualité des résultats

« On ne change pas les options de compilations car on a trouvé celles qui fonctionnent »

Un développeur de la bibliothèque

Les fausses bonnes pratiques qui cachent les erreurs

Aucune option de compilation n'est activée

1

Tous les tableaux sont globaux et alloués statiquement, la plupart sont multidimensionnels

2

Toutes les variables sont initialisées à 0

3

Rechercher les instabilités

Une identification des portions de code instables identifiée en altérant les options de compilation

L'utilisation des options de compilation les plus agressives sur l'ensemble du code ne permet pas de reproduire le comportement du code

Une mise en place progressive fichier par fichier permet d'identifier les sources d'instabilité

- 1 Des calculs effectués plusieurs fois dans des contextes différents et induisant des branchements
- 2 Calculs de différences entre des valeurs censées être calculées de façon identiques

La factorisation des calculs permet d'améliorer stabilité et performances

Afin d'éviter que certains résultats soient recalculés plusieurs fois, ils sont stockés

Certaines parties d'algorithmes sont réécrites afin d'explicitement des comparaisons sur des paramètres entiers plutôt que sur les résultats

Certaines règles métier sont explicitées afin d'éviter de calculer des valeurs connues *a priori*

Selon le contexte, on utilise un opérateur différent

On part du principe que

$a < b$ n'est pas équivalent à $b \geq a$

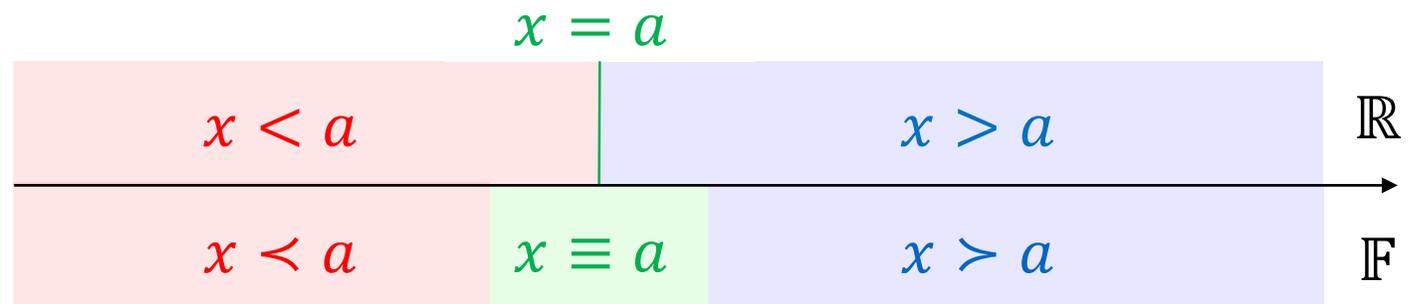
Choisir ce que l'on compare

La prise en compte de l'objectif de la comparaison conduit à une comparaison différente selon les cas

Ordonnancer des éléments

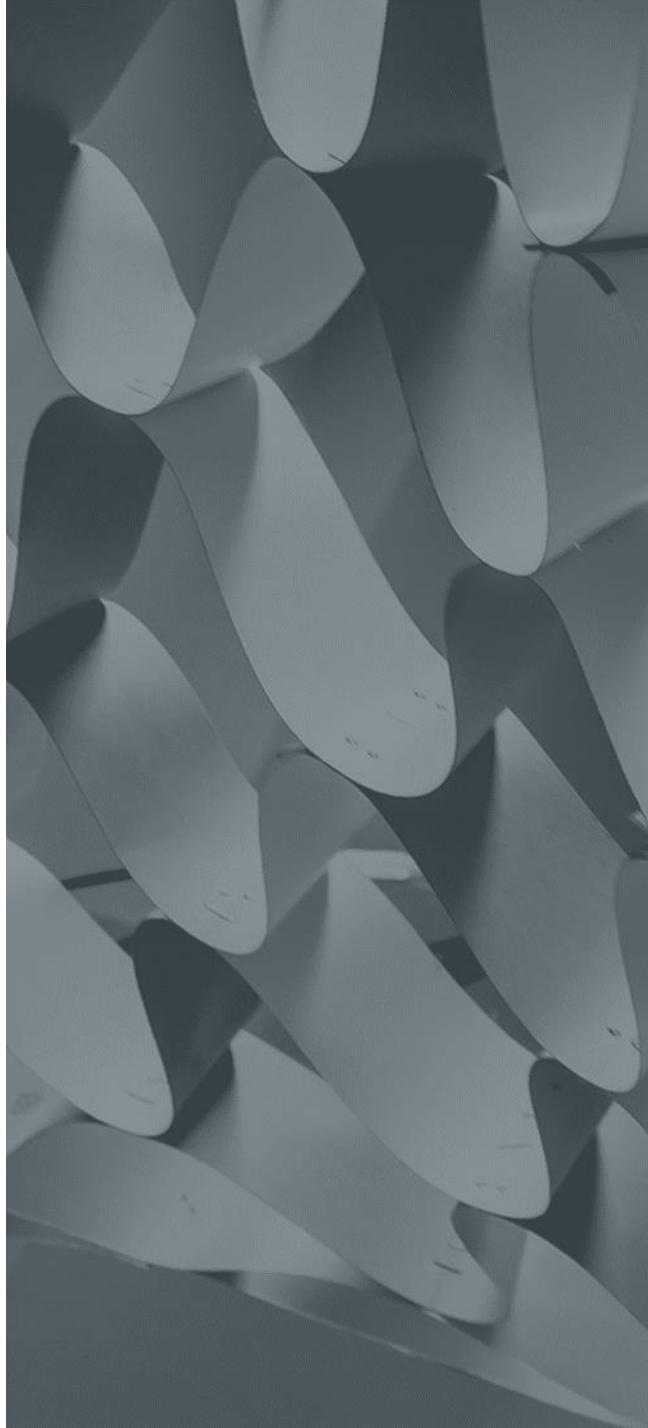
Valider l'appartenance à un domaine de définition

Définir le cas à traiter



Nos besoins pour aller plus loin

12



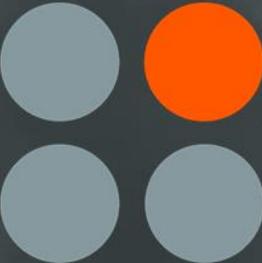
Disponibilité d'outils sous windows

Portage de VERROU s'appuyant sur DynamoRIO plutôt que Valgrind

Sensibiliser nos clients aux enjeux

Expérimentations en cours en machine learning



aneeo 

the other solution