



Référence : FOR-CP

Niveau : 2- Avancé

Contact : Virginie Pigeat

Durée : 3 jours (21h.)

Classe à distance : Possible

01 69 15 32 32 / 07 87 10 03 92

Tarif : 1 500 € h.t / personne

virginie.pigeat@agenium.com

## Objectifs

Cette formation vise à acquérir les bases de Fortran et une compréhension des architectures matérielles pour pouvoir écrire du code performant.

Connaître les réflexes pour écrire un code performant. Gagner en performance en connaissant son matériel et les contraintes qu'il impose au code.

## Public

Cette formation Fortran s'adresse aux développeurs, ayant une connaissance de base du Fortran et/ou en algorithmie.

## Pré-requis

Pour suivre cette formation, les participants doivent disposer d'une expérience de programmation en Fortran.

## Travaux pratiques

## Moyens pédagogiques et techniques

Les formations Agenium Campus sont conçues et animées par des experts en activité.

Nos salles sont équipées de vidéoprojecteur et écran/tableau et d'un accès internet. Chaque participant dispose d'un poste de travail et d'un support de cours.

Nos formations peuvent être suivies à distance.

## Modalités de suivi et d'évaluation

Les participants signent une feuille de présence par demi-journée. Une attestation de validation des acquis est remise à la fin de la formation.

L'évaluation en cours de formation est réalisée grâce à des exercices ou études de cas (50% du temps minimum pour les cours pratiques) et/ou sous forme de QCM.

L'évaluation en fin de formation Un QCM ou un exercice est donné aux stagiaires après la formation afin de mesurer l'acquisition des connaissances.

Niveau de satisfaction : 5 / 5



*Le contenu de nos formations est adaptable selon vos besoins*

## Programme :

### Module 1 – Les bases de Fortran

1. Types
  - Types de base
  - Types dérivés
2. Modules
3. Orienté objet
  - Méthodes
  - Surcharge de fonctions
4. Mémoire
  - La pile / le tas
  - Implications sur la performance
5. Allocations dynamiques
  - allocate
  - allocated
  - deallocate
  - move\_alloc
6. Pointeurs

### Module 2 – Architectures des ordinateurs

1. Architecture générale d'un processeur
  - Notions de SIMD
  - Opérations sur les nombres
    - + Représentation IEEE
    - + Nombres dénormaux
    - + Types natifs à la puce
2. Accès aux caches
  - Parcours de données
  - Row major / column major
  - Tuilage
3. Pipeline du processeur
  - Répercussion sur les boucles
    - + Utilisation de *if* dans les boucles
4. Structures de données
  - + Array of Structures
  - + Structure of Arrays

### Module 3 – Outils d'optimisation

1. Flags intéressants pour la compilation
2. VTUNE
3. GNU Perf
  - Utilisation
  - Les flamegraphs
4. Réflexes d'optimisation
  - Roofline model
    - + Utilité
    - + Utilisation
  - Boucles et branchements
    - + Fusion de boucles
    - + Fission de boucles
    - + Extraction des branchements
    - + Extraction des cas limites (bords d'une image par exemple)
  - Approximations numériques
    - + Algorithmes adaptés
    - + Précision variable
  - Organisation des données
    - + Tableau vs. table de hachage

### Module 4 – Bibliothèques de calcul

1. BLAS
  - BLAS 1, 2, 3
  - MKL
2. FFTW
3. Interfacage avec d'autres langages
  - *iso\_c\_binding*
  - Présentation des *intrinsics* utilisables en C.