



*Souvent pour prendre en main un système de simulation, on commence par des cas concrets et simples ne nécessitant pas la prise en main du code au-delà de la simple obtention des exécutables. C'est un premier pas qui permet souvent de comprendre quels sont les points d'entrées dans le code qui nous intéressent et qui méritent d'être approfondis. Du coup si votre problématique nécessite de prendre en main OpenFoam pour y programmer de nouveaux modèles, peut-être avez-vous commencé par là ?*

*Je maîtrise déjà l'usage d'OpenFoam. Je l'ai utilisé de nombreuses fois et j'ai codé mon propre solveur avec les fonctions de haut niveaux d'OpenFoam. C'est assez simple lorsqu'on ne se plonge pas dans les arcanes car la syntaxe d'OpenFoam est assez proche du langage naturelle mathématique*

Par exemple pour écrire  $\nabla(\vec{u} \otimes \vec{u}) + \nabla(\nu \nabla \vec{u}) = -\nabla p$  on écrit :

```
tmp<fvVectorMatrix> tUEqn
(
    fvm::div(phi, U)
  + fvm::laplacian(nu,U)
);
fvVectorMatrix& UEqn = tUEqn.ref();

solve(UEqn == -fvc::grad(p));
```

C'est ainsi que j'ai pu facilement répondre à mes besoins basiques. Le problème c'est de comprendre le fonctionnement interne du code qui n'est pas encore tout à fait documenté. Je n'arrive pas à le lire. Par exemple j'aimerais savoir comment est calculé un `fvc::div(U)` quand `U` est du type `volVectorField`. Je comprends bien que cela se passe dans ce fichier [https://www.openfoam.com/documentation/guides/latest/api/fvcDiv\\_8C\\_source.html](https://www.openfoam.com/documentation/guides/latest/api/fvcDiv_8C_source.html) vers les lignes 77 à 95 mais je n'arrive pas à remonter les différentes fonctions jusqu'à arriver à la somme des flux sur les faces (i.e  $\int_{\Omega} \text{div}(U) dV \approx \sum_i U_i S_i \vec{n}_i dV$ )

J'aimerais donc apprendre le C++ afin de me retrouver là dedans et pourquoi pas pouvoir coder mes propres classes de bas niveau dans OpenFoam pour mes besoins.

*De là où je suis, il me semble qu'il ne devrait rien y avoir de difficile là-dedans pour qui maîtrise déjà la programmation orientée objet en Python. À une exception près probablement : la maîtrise des pointeurs et références. Les idiosyncrasies de la syntaxe C++ rendant la chose plus complexe à bien saisir ; sachant que c'est souvent un problème pour les débutants. Sur ce sujet je recommanderai de commencer par bien comprendre la syntaxe C des pointeurs, puis de prendre un grand bol d'humilité soumise, et de se lancer dans les références C++.*

Justement, j'ai beaucoup de mal avec ces références et ces pointeurs. Je ne connais pas du tout le C. Est-ce qu'il faut vraiment passer par là ?

*Vous devriez rapidement y arriver à l'aide d'un manuel basique, et en cherchant à progresser par petites étapes plutôt qu'à franchir d'un coup l'obstacle. L'idée pour attaquer un gros problème de programmation est un peu la même qu'en modélisation physique : comment découper en éléments simples.*

Merci pour le conseil. Il faut que je me trouve une bonne référence et de bons exercices pour débiter.

## Re: Comprendre le C++ ou OpenFoam ?

Posté par [πρρ|βυ|ν|αι|υ|π|π|ε|w-ε|λ|ε|ι|δ|](#) ([site Web personnel](#)) le 30/11/21 à 09:51. Évalué à 2 (+0/-0).

Le C présente l'avantage d'une syntaxe claire pour les pointeurs : un opérateur de déréférencement (\*), et un opérateur pour obtenir des adresses (&). Beaucoup ont déjà un peu de mal avec ces deux là. Le C++ reprend l'opérateur (&) pour obtenir une syntaxe plus agréable, mais en lui attribuant une nouvelle utilisation (les références). De mon point de vue ça peut embrouiller quelqu'un qui ne maîtriserait pas encore les pointeurs. D'où l'idée de passer par le C pour cette base là.

Ceci-dit votre code n'a pas l'air de faire intensivement appel à des pointeurs, et votre problème de lecture paraît (vu de très loin) plus lié à la programmation objet (classes, héritage, templates). Du coup, il est peut-être mieux de se plonger là-dedans directement. Quitte à revenir sur les pointeurs si vous en croisez.

À mon sens n'importe quel ouvrage traitant de ces thèmes pourrait faire l'affaire. Pour chaque notion on peut se composer soi-même des exercices si l'ouvrage n'en propose pas. Un cas simple et intuitif pourrait-être de créer des classes d'objets géométriques.

--

« IRAFURORBREVIESTANIMUMREGEQUINISIPARETIMPERAT » — Odes — Horace

## Re: Comprendre le C++ ou OpenFoam ?

Posté par [moi1392](#) le 30/11/21 à 10:32. Évalué à 3 (+1/-0).

*Par exemple j'aimerais savoir comment est calculé un  $\text{fvc:div}(U)$  quand  $U$  est du type `volVectorField`.*

Juste pour ce point particulier, tu n'as pas besoin de connaître ou comprendre les arcanes des templates en C++ pour ça, tu peux simplement utiliser un débogueur et mettre un point d'arrêt sur la ligne en question, arrange toi pour y arriver avec le bon type (le débogueur te le dira) et entre simplement dans la fonction.

Au bout d'une suite plus ou moins interminable d'appels intermédiaires, tu te retrouveras dans la partie algorithmique qui devrait être relativement simple à comprendre pour toi.

## Commencer par le C++ c'est compliqué

Posté par [David Demelier](#) ([site Web personnel](#)) le 30/11/21 à 09:01. Évalué à 4 (+2/-0).

Salut,

Loin de moi l'idée de te décourager mais le C++ est clairement dans le top 3 des langages les plus compliqués à maîtriser au monde.

Bien sûr tu peux rapidement faire des choses simple, mais c'est un langage tellement vaste rempli de pièges et de fonctionnalités obsolètes à connaître.

À l'heure actuelle je ne connais pas de ressources correctes qui permettent d'apprendre correctement le C++20 (dernière révision). Celui qui a été le plus recommandé est [C++ primer](#) de Lippman mais il couvre que le C++11.

Dans tous les cas, suis un cours **à jour** et surtout **pas** celui du site du zero. Je pense qu'il doit y avoir des cours en ligne, il faudrait que ça couvre au minimum C++17.

--

l'azerty est aux dispositions ce que subversion est aux SCMs

## Re: Commencer par le C++ c'est compliqué

Posté par [אריאל וינגר \(site Web personnel\)](#) le 30/11/21 à 10:45. Évalué à 2 (+0/-0).

« Loin de moi l'idée de te décourager mais le C++ est clairement dans le top 3 des langages les plus compliqués à maîtriser au monde. »

Je souscris totalement. Les dépêches sur C++ m'en ont totalement convaincu. Ceci dit tempérons le propos. En physique nous avons heureusement pas mal de trains de retard en matière de programmation complexe. À la fois faute de temps à consacrer aux nouveautés de ce domaine connexe, et parce que nos usages sont souvent assez élémentaires. La plupart des codes de physique se lisent à partir de notions basiques de programmation, éventuellement matinées de parallélisme.

--

« IRAFURORBREVISESTANIMUMREGEQUINISIPARETIMPERAT » — Odes — Horace

## ressources

Posté par [Gil Cot](#) le 30/11/21 à 09:06. Évalué à 4 (+2/-0).

Le wiki a une page qui devrait être un passage obligé pour se lancer : [https://openfoamwiki.net/index.php/Tutorials/Before\\_Getting\\_Started](https://openfoamwiki.net/index.php/Tutorials/Before_Getting_Started) et <https://wiki.openfoam.com/Tutorials> (en particulier le « 1st glimpse » et le « 3 weeks »)

On y parle de l'installation et de la prise en main, nécessaires avant de bidouiller le code source. Le forum a un bon complément à partir de : <https://www.cfd-online.com/Forums/openfoam/146697-getting-started-openfoam.html>

Si tu préfères une formation un peu plus classique mais en ligne, tu peux regarder par exemple <https://www.udemy.com/course/openfoam-for-beginners/>

Enfin, <https://holzmann-cfd.com/community/learn-openfoam> après avoir fait l'historique et les diverses considérations, propose des liens vers [le manuel PDF de programmation](#) Je pense qu'en plus de l'utilisation du produit, il faut d'abord se frotter un peu à C++ et en ligne il y a quelques classiques : <https://www.learn-cpp.org/> et <https://www.learncpp.com/> ; ou encore <https://www.w3schools.com/cpp/default.asp> et <https://www.tutorialspoint.com/cplusplus/index.htm> et <https://www.programiz.com/cpp-programming>

En formation à distance classique, on peut regarder par exemple <https://openclassrooms.com/fr/courses/1894236-programmez-avec-le-langage-c>

Je laisse d'autres compléter (liens et retours d'expérience.)

--

"It is seldom that liberty of any kind is lost all at once." — David Hume

**Note** : les commentaires appartiennent à ceux qui les ont postés. Nous n'en sommes pas responsables.