각 step의 beginnerFOAM의 main 소스 파일(beginnerFoam_*.cpp)을 컴파일하시면 실행 파일을 생성할 수 있으며, 리눅스 g++의 경우 run.cmd를 실행하시면 모든 단계의 컴파일을 수행할 수 있습니다. Step 4까지는 별도의 입력 파일이 필요 없으나, Step 5부터 phi라는 파일이 필요하므로 각 단계에서 필요한 입력 파일은 phi.org.*를 phi 이름으로 복사하여 사용하시면 됩니다.

Step 1. c 언어에서 기본으로 제공하는 배열을 이용하여 미분 방정식을 풀게 되는 전형적인 소스 코드를 설명

Step 2. OpenFOAM의 가장 근간이 되는 UList와 List class 기능을 List class 하나로 합쳐서 배열을 대체하는 코드를 설명

Step 3. 행렬 구성 및 해석 기능을 갖는 fvMatrix class를 설명. Ax=B라는 형식에서 A라는 계수 행렬과 B라는 소스 행렬을 구성하고 해석하는 코드를 설명

Step 4. Step 3에서 경계 조건이 fvMatrix에 포함되어 있으나, 원래 경계 조건은 지배 방정식 단계에서 변수에 의해 설정되는 것이 합리적임. 따라서 물리량이 저장되는 List 기능을 상속함과 동시에 경계 조건 기능을 갖는 GeometricField class를 만들고, 격자에 대한 기본적인 기능을 하는 fvMesh class를 간단하게 만든 부분을 설명

Step 5. 물리량의 초기화를 위해 어떤 파일을 읽어들일 것인가를 결정하는 IOobject class를 설명하고 읽어 들이는 기능을 GeometricField에 구성함

Step 6. fvMatrix에서 행렬 기반(Ax=B)이 아닌 지배 방정식 기반(시간 미분, 공간 미분 등)으로 행렬을 구성하는 기능을 구현함. 단 이 때의 전제 조건은 시간과 공간은 서로 독립 변수이므로 시간 미분과 공간 미분 역시 독립된 차분 계수 행렬을 구성하는 것이 합리적이나, 2차 CrankNicolson 차분 기법은 시간 차분 정확도를 높이면서 공간 차분의 계수가 달라지는 특성이 있음. backward나 Euler의 경우에는 전혀 문제가 없으며, CrankNicolson에 대해 트릭을 사용하고 있는데 이 부분은 현재 beginnerFOAM에서 생략되어 있음

Step 7. fvMatrix에서 지배 방정식의 표현을 흉내내도록 fvm namespace와 tmp class를 설명하고

있음

Step 8. 외재적으로 값이 정해지는 Dirichlet 경계 조건은 행렬의 소스항에 관계되어 있으며, 내재적으로 값이 정해지는 Neumann 경계 조건은 행렬의 계수항과 관계되어 있으므로 경계 조건의 종류에 따라 소스항과 계수항을 바꾸는 기능을 fvMatrix에 추가함

Step 9. 경계 조건에서 기능과 값을 부과하는 Boundary class를 구현하였으며, 물리적 의미에 따라 GeometricField 내에 위치하도록 함

Step 10. 1차원 문제가 아니라 2, 3차원으로 확장될 경우 각 경계에서의 값을 부과하는 방식에 대해 고민하였으며, PtrList를 구현하는 것이 합리적임을 설명함

Step 11. Dirichlet 경계 조건에 기반한 fixedValueFvPatchField와 Neumann 경계 조건에 기반한 zeroGradientFvPatchField를 구현함

Step 12. 각 경계 조건 클래스의 기능을 보완함

Step 13. 각 경계 조건의 상위 클래스인 fvPatchField를 구현함

Step 14. OpenFOAM에서 사용하는 기능은 evaluate를 구현하고 updateCoeffs를 virtual 형태로 구현함

Step 15. run-time selection 기능을 구현하기 위해 pointer to function을 이해하고 사용할 수 있도록 기본적인 코드 업데이트 순서를 제공함. Step 15는 OpenFOAM에서 run-time selection의 구조를 이해하고자 하는 분들에게 도움이 되도록 제공되었으므로 굳이 이 부분을 이해하지 않더라도 OpenFOAM을 개발하고 사용하는 것은 별 문제가 없을 것입니다. OpenFOAM에서 run-time selection의 기본 구조를 변경하거나 좀 더 직접적으로 메모리를 통제하고자 하는 분들이 계시다면 도움이 될 수 있을 것입니다.