문제 4 (대학원): 멀티그리드 및 Conjugate Gradient 방법 기반의 Poisson PDE 계산 병렬화 (배점: 대학원 30 점)

문제 개요

Poisson 방정식은 전기전자/재료/물리/화학/기계 등의 매우 다양한 계산과학분야에서 폭넓게 활용되는 2차 편미분방정식 (Partial Differential Equation, PDE) 중 하나로, 주어진 조건 ρ에 대한 해 Ψ를 구하는 것이 그 목적이다. 실함수 Ψ가 유클리드 공간 에서 두번 미분가능한 경우 Poisson 방정식은 수식 (1)과 같이 정의된다. 2차원 공간 에서 Ψ 와 ρ 는 공간변수 x와 y의 함수가 되며 (Ψ(x, y), ρ(x, y)), Poisson 방정식은 수식 (2)와 같다.

$$\nabla^2\Psi = \rho \qquad (1)$$

$$\frac{d^2\Psi(x,y)}{dx^2} + \frac{d^2\Psi(x,y)}{dx^2} = \rho(x,y) \qquad (2)$$

유한차분법(Finite Difference Method)을 이용해 컴퓨터로 수식 (2)의 Poisson 방정식을 푸는 경우, 수식 (2)의 Poisson 방정식은 결국 선형시스템의 해를 계산하는 문제가 되며, 수식 (3)에 보여진 것과 같이 정방형 행렬 A와 RHS (Right-Hand-Side) 벡터 b가 주어 졌을 때의 해 z를 구하는 문제와 같아진다.

$$Az = b \qquad (3)$$

문제 설명

2차원 공간 (x, y) 에서 수식 (4)와 같이 정의되고, 수식 (5)의 경계조건을 만족하는 Poisson 방정식의 해 Ψ(x, y)를 그림 1의 CG 알고리즘과 문제 2에서 설명된 멀티그리드 방법을 이용하여 계산하는 순차코드가 C 와 Fortran 으로 주어져 있다.

$$\frac{d^2\Psi(x,y)}{dx^2} + \frac{d^2\Psi(x,y)}{dx^2} = \sin(\pi x)\sin(\pi y) \quad (0 \le \mathrm{x,y} \le 1) \qquad (4)$$

$$\Psi(x,0) = \Psi(0,y) = \Psi(x,1) = \Psi(1,y) = 0 \qquad (5)$$

본 문제에서는 방정식 (3)을 풀기 위해 2번 문제에서 설명된 멀티그리드 방법을 이용한다. Restriction 단계에서는 방정식 (3)의 우변 (RHS, right-handed side)을 fine grid에서 coarse grid로 연속적으로 restriction하여 coarsest grid에서의 RHS를 만들고 이를 풀어 coarsest grid에서의 해를 구한다. 이후에는 coarse grid의 해를 interpolation하여 fine grid의 초기 조건으로 이용하며, 각 grid에서 연속적으로 interpolation과 해를 구하는 solving 과정을 반복하여 최종적으로 원래 grid에서의 해를 계산한다. (그림 2 참고)

주어진 Poisson 방정식 계산 순차코드를 병렬화하되, 다음에 제시된 병렬화 과정에서의 유의사항을 위반하는 경우 제출한 코드의 성능과 상관없이 0점으로 평가됨에 유의한다.

- **A**: (NxN) matrix.
- **x, r, b, p**: (Nx1) vector.

We want to solve $\mathbf{Ax} = \mathbf{b}$. First compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$, $\mathbf{p}_0 = \mathbf{r}_0$

loop for (j=1; j<=**K** ; j++)

  $\mathbf{a}_j \leftarrow \langle \mathbf{r}_j \bullet \mathbf{r}_j \rangle / \langle \mathbf{Ap}_j \bullet \mathbf{p}_j \rangle$;

  $\mathbf{x}_{j+1} \leftarrow \mathbf{x}_j + \mathbf{a}_j \mathbf{p}_j$;

  $\mathbf{r}_{j+1} \leftarrow \mathbf{r}_j - \mathbf{a}_j \mathbf{Ap}_j$;

  if ($\|\mathbf{r}_{j+1}\| / \|\mathbf{r}_0\| < \mathbf{e}$)

    declare $\mathbf{r}_{j+1}$ is the solution of $\mathbf{Ax} = \mathbf{b}$ and break the loop

  $\mathbf{c}_j \leftarrow \langle \mathbf{r}_{j+1} \bullet \mathbf{r}_{j+1} \rangle / \langle \mathbf{r}_j \bullet \mathbf{r}_j \rangle$;

  $\mathbf{p}_{j+1} \leftarrow \mathbf{r}_{j+1} + \mathbf{c}_j \mathbf{p}_j$;

end loop

그림 1. Conjugate Gradient 알고리즘의 실행 Flow

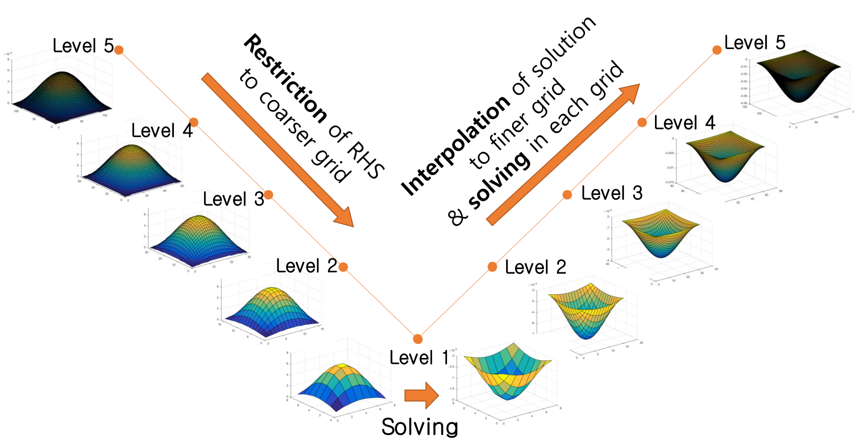

그림 2. 멀티그리드 알고리즘의 실행 Flow

참고 및 유의 사항

1. 주어진 순차코드는 문제 2에서 설명된 (3개) 계층의 격자를 가지는 V-cycle 멀티그리드 방법과 그림 1에 제시된 Conjugate Gradient (CG) 알고리즘을 이용해 Poisson 방정식의 해를 계산한다. 제시된 CG 알고리즘과 멀티그리드 알고리즘의 실행 Flow 및 주어진 순차코드의 subroutine 호출구조를 유지하는 범위에서의 코드변경은 허용하나, **CG** 외의 다른 알고리즘이나 공개코드를 사용해 주어진 subroutine 의 호출구조를 변경하는 행위는 금지한다. 또한 주어진 **restriction 및 interpolation**외의 다른 알고리즘/공개코드를 사용해 주어진 subroutine의 호출구조를 변경하는 행위 또한 금지한다.
2. 주어진 순차코드는 poisson.c (f90) 에서 cgsolver.c (f90) 와 matrix_constructor.c (f90) 에 정의된 CG 관련 서브루틴과, multigrid.c (f90)에 정의된 멀티그리드 관련 서브 루틴들을 호출해 사용하는 방식으로 수행된다. 컴파일과 링크를 위한 Makefile 이 함께 주어져 있으며 이를 변경하는 것은 금지한다.
3. 동적메모리 할당/삭제는 반드시 시간측정 구간 안에서 수행한다. 순차코드에 정의된 차분계수 (gridsize), 수렴조건 (tolerance), iteration 한도 (maxiteration), 레벨 수 (nlevels) 의 변경은 금지한다.
4. 본 문제는 편미분방정식, 유한차분법, 멀티그리드, CG 알고리즘의 사전지식이 없어도 풀 수 있다는 점을 참고한다. 행렬로 구성된 선형시스템을 푸는 문제이고, 순차 코드에 행렬 및 RHS 의 모든 정보가 주어져 있다. 주어진 순차코드의 해를 위해 그림 1에 제시된 CG 알고리즘의 실행 Flow와 그림 2에 주어진 멀티그리드 알고리즘의 실행 Flow를 참고한다. 벡터의 내적, 행렬-벡터의 곱만 알면 해석 가능한 알고리즘이다.

평가 방법

1. 병렬코드가 32 core 에서 수행되는 시간을 time command 로 측정한다.
2. 순차코드 실행이 끝나면 result 폴더 밑에 solution.dat 파일이 생성된다. 병렬 코드는 순차코드의 해와 오차범위내 (1e-6) 에서 같은 해를 생성해야 하며, 해의 계산에 소요되는 iteration 수는 정확히 일치해야 한다. 본 조건이 만족되지 않는 경우 코드의 정확성 미달로 결격 처리한다.