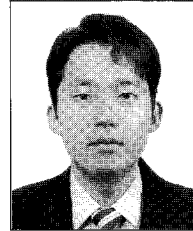
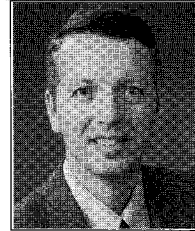


일반유한요소법을 이용한 3차원 균열의 효율적인 병렬연산 시뮬레이션

Efficient Parallel Simulations of Three-Dimensional Cracks Using the Generalized Finite Element Method



김 대 진*



C. Armando Duarte**

* 경기대학교 건축공학과 조교수
** 일리노이주립대(UIUC) 토목환경공학과 부교수

1. 서론

파괴역학(fracture mechanics) 문제의 현실적인 시뮬레이션은 유한요소법을 사용하더라도 여전히 어려운 과제이다. 3차원 파괴역학 문제의 정확한 해를 수치해석적으로 구하기 위해서는 균열 선단 주변의 요소망을 세분화할 뿐 아니라 높은 차수의 형상함수를 사용하여야 한다. 이는 병렬연산기법을 이용한 해석에 있어서도 많은 어려움을 야기하는데 각각의 프로세서에 계산량을 균등하게 할당시키는 것이 쉽지 않기 때문이다. 일반적인 병렬연산 유한요소 기법은 각각의 프로세서가 동일한 계산 부하를 갖도록 문제의 영역을 적절히 분할하도록 하고 있다. 하지만 균등하지 않은 요소망 및 형상함수 차수를 지닌 유한요소법 이산화의 경우 각각의 분할된 영역에 대한 계산량의 예측이 쉽지 않아(Oden et al., 1992) 일반적인 병렬연산기법의 적용에 어려움이 있다. 뿐만 아니라 균열 전파 해석의 경우 시뮬레이션 도중 각각의 분할된 영역의 계산량이 변화하기 때문에, 단순히 많은 양의 계산을 수행할 수 있는 슈퍼컴퓨터를 사용하는 것만으로 충분하지 않다. 대신 기존의 기법을 개선하여 이러한 종류의 문제를 병렬연산을 통해 효율적으로 풀 수 있는 수치해석 알고리즘의 개발이 필요하다.

일반 혹은 확장유한요소법(generalized/extended finite element method)은 다항식이 아닌 형상함수를 사용할 뿐만 아니라 불연속면(discontinuity) 혹은 다물질 경계면(multi-material interface)와 같은 해에 대한 사전 정보(apriori knowledge)를 유한요소의 이산화 해공간에 확장함수(enrichment function)의 형태로 적용시키기 위해 개발되었다(Oden et al., 1998; Moes et al., 1999; Duarte et al., 2000). 그러나 이러한 사전 정보는 일반적으로 수학적인 정해의 형태로 주어지기 때문에 많은 경우에 있어 일반/확장유한요소법의 적용이 2차원 선형탄성문제에 제한되어져 왔다. 이에 Duarte와 Kim 등은 확장함수를 수치 해석적인 방법으로 구성하기 위한 전체-국부 확장함수를 지닌 일반유한요소법을 제안하였다(Duarte and Kim, 2008, Kim, Duarte and Pereira, 2010). 이 기법은 상호 의존적인 전체(global) 및 국부(local) 스케일 문제의 해에 기반하고 있다. 국부문제(local problem)는 3차원 균열 주변의 미세 스케일(fine scale) 거동을 묘사하는데 초점을 맞추며 전체문제(global problem)는 거시 스케일(macro scale)의 거동을 다룬다. 국부문제의 해는 단위오목분할(partition of unity)의 개념을 통해 확장함수의 형태로 전체 문제의 해공간을 확장하는데 이용된다. 본 기사에서는 이 기법을 이용하여 3차원 균열 문제를 해석할 경우 대부분의 해석시간이 소요되는 국부문제의 해를 병렬연산을 활용하

여 구하는 방법을 소개한다. 이러한 국부문제의 해는 병렬 연산 기법을 이용해 매우 효율적으로 구해질 수 있는데 프로세서 사이에 아무런 교신(communication)이 필요하지 않으며 각각의 프로세서에 계산량을 균등하게 할당시킬 수 있기 때문이다. 수치해석 예제를 통해 소개된 기법의 정확성과 병렬연산의 효율성을 입증한다.

2. 전체-국부 확장함수를 지닌 일반유한 요소법

본 장에서는 전체-국부 확장함수를 지닌 일반유한요소법(generalized finite element method with global-local enrichment functions)의 자세한 수식을 소개한다. 이 기법은 성긴 요소망을 이용해 구해진 전체문제의 해를 국부문제의 경계 조건으로 이용하며 국부문제는 요소망의 세분화나 높은 차수의 형상함수를 사용함으로써 정확한 해를 구하게 된다. 국부문제의 해는 다시 확장함수의 형태로 전체문제의 해공간을 확장하는데 사용되며, 이는 적은 자유도를 이용하여 정확한 해를 구하는 것을 가능하게 한다. 이후로 편의상 이 기법을 GFEMg-I로 명명한다.

2.1 초기 전체문제

힘의 평형을 묘사하는 편미분방정식은 3차원 영역(domain) $\Omega_G = \Omega_G \cup \partial\Omega_G$ 에 정의되며, 이의 경계는 $\partial\Omega_G^u \cap \partial\Omega_G^t = 0$ 을 만족시키는 두 가지 영역의 합 $\partial\Omega_G = \partial\Omega_G^u \cup \partial\Omega_G^t$ 으로 구성된다. 영역 Ω_G 에 정의된 힘의 평형방정식과 구성방정식은

$$\nabla \cdot \sigma = 0, \sigma = C : \epsilon \quad (1)$$

으로 정의되며, 경계 조건은 $\partial\Omega_G$ 에

$$u = \bar{u} \text{ on } \partial\Omega_G^u, \sigma \cdot n = \bar{t} \text{ on } \partial\Omega_G^t \quad (2)$$

으로 주어진다. C 는 Hooke 텐서, n 은 $\partial\Omega_G$ 에 직교하는 바깥 방향으로의 단위 벡터이며, \bar{t} 와 \bar{u} 는 경계치 문제의 주어진 견인력(traction)과 변위(displacement)이다. u_G^0 는 식 (1)과 (2)에 의해 정의되는 문제의 일반유한요소해로 다음의 적분식으로부터 구한다:

$\forall v_G^0 \in X_G^0(\Omega_G)$ 에 대해 $u_G^0 \in X_G^0(\Omega_G) \subset H^1(\Omega_G)$ 을 만족시키는 다음 적분식의 해 u_G^0 를 구하라.

$$\int_{\Omega_G} \sigma(u_G^0) : \epsilon(v_G^0) dx + \eta \int_{\partial\Omega_G^t} u_G^0 \cdot v_G^0 ds = \int_{\partial\Omega_G^t} \bar{t} \cdot v_G^0 ds + \eta \int_{\partial\Omega_G^u} \bar{u} \cdot v_G^0 ds \quad (3)$$

$X_G^0(\Omega_G)$ 는 Ω_G 에 정의된 힐베르트 해 공간 $H^1(\Omega_G)$ 의 일반유한요소 형상함수를 이용한 이산화(discretization)이다. 식 (3)의 변수 η 는 벌칙계수(penalty parameter)이다. 식 (3)에 의해 정의된 문제를 풀기 위해 일반적으로 성긴 유한요소망을 사용하며 일반적으로 미세 균열과 같은 국부적인 특징은 정의되지 않는다. 편의상 이 문제를 선형 초기 전체문제(linear initial global problem)라 명명한다.

2.2 국부문제

Ω_L 을 균열 선단(crack front) 주변과 같이 국부적인 복잡한 거동을 보이는 Ω_G 의 하부영역(subdomain)으로 정의하고 다음의 국부문제의 해 u_L 를 구한다:

$\forall v_L \in X_L(\Omega_L)$ 에 대해 $u_L \in X_L(\Omega_L) \subset H^1(\Omega_L)$ 을 만족시키는 다음 적분식의 해 u_L 를 구하라.

$$\int_{\Omega_G} \sigma(u_L) : \epsilon(v_L) dx + \eta \int_{\partial\Omega_L \cap \partial\Omega_G^t} u_L \cdot v_L ds + \kappa \int_{\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)} u_L \cdot v_L ds = \int_{\partial\Omega_L \cap \partial\Omega_G^t} \bar{t} \cdot v_L ds + \eta \int_{\partial\Omega_L \cap \partial\Omega_G^u} \bar{u} \cdot v_L ds + \int_{\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)} (t(u_G^0) + \kappa u_G^0) \cdot v_L ds \quad (4)$$

$X_L(\Omega_L)$ 는 Ω_L 에 정의된 힐베르트 해 공간 $H^1(\Omega_L)$ 의 일반유한요소 형상함수를 이용한 이산화(discretization)이다. 이 문제의 핵심적인 사항은 초기 전체문제의 해 u_G^0 를 $\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)$ 에서의 경계조건으로 이용하고, 이 문제의 해 u_L 는 영역 Ω_L 에서의 국부적인 거동을 정확하게 묘사할 수 있다는 것이다. 변수 η 는 벌칙계수(penalty parameter)이며 스프링 강성 κ 은 (Kim, Duarte and Pereira, 2010)에 소개된 방식에 의해 선택할 수 있다. 이 문제를 편의상 국부문제(local problem)라 명명한다.

2.3 전체-국부 확장함수와 확장 전체문제

2.1장과 2.2장에 묘사된 초기 전체문제와 국부문제는 전

체문제에 전체-국부 확장함수 u_L 를 공급하기 위한 사전 과 정이다. u_L 에 의해 확장된 전체-국부 일반유한요소 형상 함 수는 다음과 같이 표현된다.

$$\phi_\alpha(x) = \varphi_\alpha(x)u_L(x) \quad (5)$$

φ_α 는 전체문제의 절점 α 에 정의된 단위 오목 분할 함수 (partition of unity function)이며 선형 Lagrangian 유한요소 형상함수를 이용한다. u_L 은 전체-국부 확장함수(global-local enrichment function)이며, 이 함수에 의해 확장된 전체문제는 확장 전체문제(enriched global problem)이라 명명된다. 이 문제의 약식(weak formulation)은 다음과 같이 주어진다.

$\forall v_G^E \in X_G^E(\Omega_G)$ 에 대해 $u_G^E \in X_G^E(\Omega_G) \subset H^1(\Omega_G)$ 을 만족 시키는 다음 적분식의 해 u_G^E 를 구하라.

$$\int_{\Omega_G} \sigma(u_G^E) : \epsilon(v_G^E) dx + \eta \int_{\partial\Omega_G} u_G^E \cdot v_G^E ds = \int_{\partial\Omega_G} t \cdot v_G^E ds + \eta \int_{\partial\Omega_G} \bar{u} \cdot v_G^E ds \quad (6)$$

$X_G^E(\Omega_G)$ 는 $X_G^0(\Omega_G)$ 에 식 (5)로 정의된 일반유한요소 형 상함수가 Ω_G 에 추가된 힐베르트 해 공간 $H^1(\Omega_L)$ 의 이산 화(discretization)이다. 전체-국부 확장함수는 3차원 탄성 문 제를 해석할 경우 국부문제의 자유도의 개수와 상관없이 성긴 전체문제의 요소망에 절점 당 오직 3개의 자유도만이 계층적으로(hierarchically) 추가된다. 따라서 전체문제의 크 기를 크게 증가시키지 않으면서도 주어진 문제의 미소 스 케일 거동을 묘사하기 위해 국부문제에 세분화된 요소망을 사용하거나 높은 차수의 형상 함수를 사용하는 것이 가능하 며, 균열 등과 같은 국부적인 특징을 포함하는 문제를 해석 하기 위해 확장 전체문제에 추가되는 자유도의 개수는 일반 적으로 전체 자유도 개수에 비해 훨씬 적다. 전체-국부 확장 함수의 계층적인 추가로 인해 확장 전체문제는 Duarte 등 (2008)에 소개된 정적 압축법(static condensation)을 이용해 효율적으로 풀 수 있다. Kim 등(2010)에 보인 것처럼 대부 분의 실용적인 공학 애플리케이션에 있어 이 기법을 활용 해 확장 전체문제를 푸는데 필요한 계산량은 그다지 많지 않다. 따라서 본 원고에서는 국부문제의 병렬 연산해의 정 확성과 효율성에 초점을 맞춘다.

3. 전체-국부 확장함수의 병렬 연산

본 장에서는 국부문제를 병렬연산으로 풀기위한 몇몇 기

술적인 이슈에 대해 논의한다. 이후로 편의상 이 기법을 병 렬연산 GFEMg-1로 명명한다.

3.1 마스터-서브 국부문제 기법

병렬연산 GFEMg-1 기법의 핵심적인 아이디어는 병렬연 산의 효율성(parallel efficiency)을 높이기 위해 국부문제가 정의되는 영역 Ω_L 에 많은 수의 국부문제를 정의하는 것이 다. 이 기법에서 가능한 가장 작은 크기의 국부문제는 전체 영역의 절점 α 에 정의되는 하나의 클라우드 혹은 패치 (cloud or patch, 절점을 공유하는 요소들의 집합)이며 이를 국부문제의 영역으로 정의한다. 이 방법으로 국부문제를 정의할 때 하나의 전체 요소(global element)가 서로 다른 두 개 이상의 국부문제의 해로 확장될 수 있다. 국부문제의 해로 확장된 전체 요소의 수치적분이 국부문제의 요소망에 근거하여 수행되기 때문에 이들 두 개 이상의 국부문제를 구성하는 요소망이 동일하지 않을 경우 수치 적분을 위한 국부문제의 요소망의 선택에 어려움이 있다. 따라서 이를 해결하기 위해 마스터-서브 국부문제 기법(master-sub local problem approach)이 Kim 등(2011)에 의해 제안되었다.

제안된 마스터-서브 국부문제 기법의 개념이 그림 1에 나타나 있다. 국부문제를 전체 요소망으로부터 바로 생성 하는 대신 우선 마스터 국부 영역(master local domain)을 생성시킨다. 정확한 국부문제의 해를 얻기 위해 이 영역에 hp-적응(adaptation)이 실시되며, 성긴 전체 요소망으로부터 의 클라우드가 hp-적응된 마스터 국부문제로부터 서브 국 부문제(sub local problem)를 추출하기 위해 이용된다. 서브 국부문제의 해는 병렬연산에 의해 구해지며 이들은 전체문 제의 해를 확장하는데 이용된다. 전체 요소망 내의 각각의 클라우드는 하나의 서브 국부문제를 정의하기 위해 이용된 다. 이와 같은 방법을 통해 많은 수의 서브 국부문제가 생 성될 수 있으며, 이는 각각의 프로세서에 할당되는 부하의 균등화(load balancing)에 유리하다.

3.2 병렬연산 알고리즘

병렬연산 GFEMg-1의 주된 장점은 국부문제의 해를 병렬 연산을 통해 구하는 과정에서 프로세서 사이에 어떠한 교 신도 필요하지 않다는 것이다. 각각의 서브 국부문제를 풀 기 위해 필요한 모든 정보는 병렬연산을 실시하기 이전 초 기 전체문제의 해로부터 얻어지며, 구해진 각각의 서브 국 부문제의 병렬연산 해는 해공간의 확장을 위해 전체문제로 보내진다. 기법의 이와 같은 특징은 병렬연산의 구현을

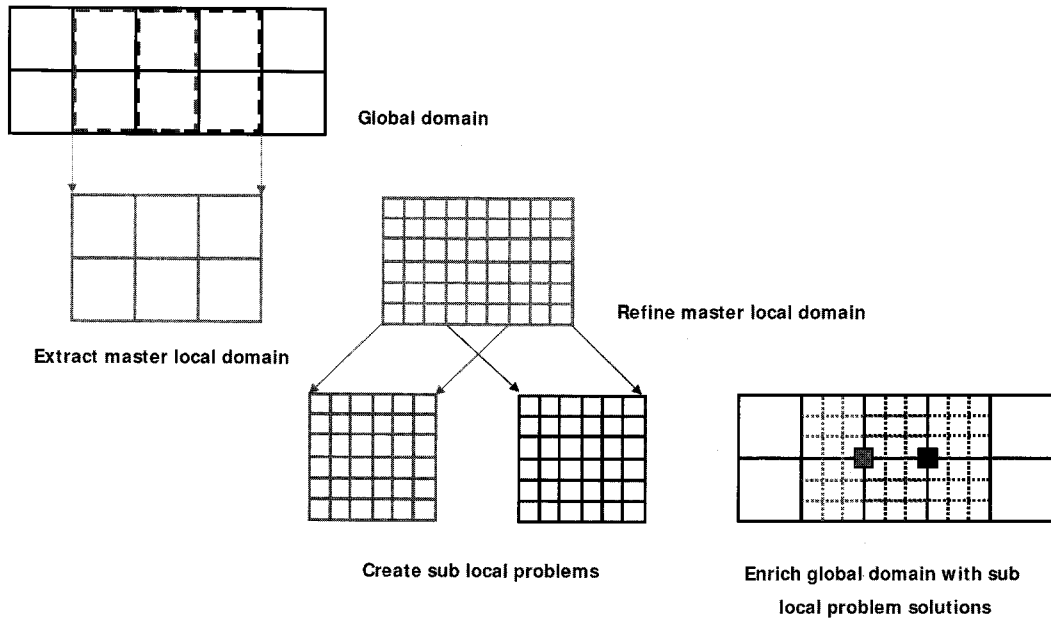


그림 1 마스터-서브 국부문제 기법(Kim et al., 2011)

```

Create sub-local problems using the master-sub local problem approach and store them in a list;
Sort sub-local problems in the list;
Activate  $N_p$  processors;
foreach processor  $i, i = 1, \dots, N_p$  do
    while not all of the sub-local problems in the list are solved do
        while the OpenMP lock is not available do
            Wait until the OpenMP lock becomes available;
        end while
        set an OpenMP lock;
        select the first sub-local problem in the list and remove it from the list;
        unset the OpenMP lock;
        solve the selected sub-local problem;
    end while
end foreach
    
```

Figure 6: Parallel algorithm for the solution of sub-local problems implemented using OpenMP. A critical region can be set using the OpenMP lock and only one processor is allowed to enter into the critical region.

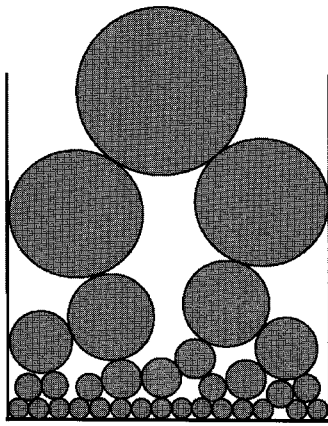
그림 2 OpenMP를 이용한 병렬연산기법으로 서브-국부문제의 해를 구하는 알고리즘(Kim et al., 2011)

극도로 단순화시키며 이는 OpenMP(Open Multi-Processing) 프로그래밍 모델을 이용해 이루어진다. OpenMP 모델은 메모리를 공유하는 멀티-프로세서 플랫폼(multi-processor platform)에서 병렬연산을 수행하는데 적합하다. 병렬연산 GFEMg-1의 알고리즘은 다음과 같이 요약되며 그림 2에 도식화되어 있다.

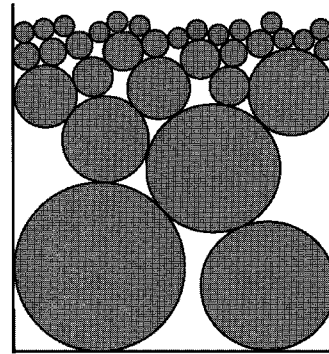
(i) 3.1장에 묘사된 마스터-서브 국부문제 기법을 이용해 서브 국부문제를 생성시켜 이들의 하나의 리스

트에 저장한다.

- (ii) 리스트 안의 각각의 서브 국부문제를 계산량의 예측치에 근거하여 분류한다. 분류 기준은 3.3장에 논의된다.
- (iii) 사용자에게 의해 정의된 N_p 개의 프로세서를 활성화시킨다.
- (iv) 리스트 안의 첫 번째 서브 국부문제를 선택하고 리스트로부터 제외시킨다. 이는 임계영역(critical region)에서의 소위 경쟁조건(race condition)을 피하기 위해



(a) Sorting in ascending order



(b) Sorting in descending order

그림 3 내림 및 오름 차순으로 서브-국부문제의 분류(Kim et al. 2011)

필요하다. 오직 하나의 프로세서만이 임계영역에 들어가는 것이 허락된다.

- (v) 각각의 서브 국부문제의 해를 병렬연산으로 구한다. 대부분의 CPU 시간이 이 단계에서 소요된다.
- (vi) 리스트 안의 모든 서브 국부문제가 풀리지 않은 경우 (iv) 단계로 되돌아가 과정을 반복한다. 그렇지 않은 경우 서브 국부문제의 병렬연산을 종료한다.

3.3 서브 국부문제의 분류

국부적인 요소망의 세분화 혹은 p-확장(p-enrichment)이 수행될 경우 각각의 서브 국부문제를 풀기 위한 계산량은 크게 차이가 날 수 있다. 이 경우 앞서 소개된 알고리즘을 이용하더라도 각각의 프로세서에 할당되는 계산량이 불균등하게 분배될 수 있으며, 이는 병렬연산의 효율성을 저하시킨다. 따라서 그림 2에 언급된 알고리즘에 이용된 리스트 내의 서브 국부문제들을 크기에 따라 분류시킴으로써 계산량의 불균등 문제를 해결한다. 그림 3(a)는 리스트 내의 서브 국부문제들에 대한 가능한 최악의 분류를 보여준다. 이 그림에서 각각의 서브 국부문제는 원에 의해 표현되며 서브 국부문제를 풀기 위한 계산량은 원의 크기로 대표된다. 그림에서 서브 국부문제들은 오름 차순으로 분류되어 있는데 - 가장 작은 문제가 가장 먼저, 가장 큰 문제가 가장 나중에 풀린다 - 각각의 서브 국부문제의 계산량의 차이가 클 경우 프로세서 사이에 심각한 계산량 불균등 문제가 발생하게 된다. 그림 3(b)는 반대의 경우를 보여준다. 리스트 내의 서브 국부문제들은 내림 차순으로 분류되어 있으며 - 가장 큰 문제가 가장 먼저, 가장 작은 문제가 가장 나

중에 풀린다 - 프로세서 간의 계산량 불균등 문제를 최소화시킬 수 있다. 아무런 분류가 행해지지 않는 경우 병렬연산의 효율성은 설명된 두 극단적인 상황의 중간에 위치하게 된다.

4. 수치 해석 예제

본 장에서는 앞서 소개한 병렬연산 GFEMg-1의 정확성과 효율성을 입증하기 위한 수치 해석 예제 두 개를 소개한다.

4.1 정육면체 도메인에 포함된 원형균열

첫 번째 수치 해석 예제로 그림 4에 나타난 원형 균열을 제안된 병렬연산 GFEMg-1을 이용해 해석한다. 크기 σ 를 가진 y축 방향으로의 인장 견인력(tensile traction)이 정육면체 도메인의 상부와 하부에 작용하며 다음과 같은 변수를 이용한다: 도메인 사이즈 $2L = 2.0$; 균열 반지름 $a = 0.5$; 수직방향 견인력 $\sigma = 1.0$; 탄성계수 $E = 2.0 \times 10^5$; 포아송 계수 $\nu = 0.3$.

그림 4에 나타난 것처럼 전체문제는 $6 \times (10 \times 10 \times 10)$ 개의 사면체(tetrahedron) 요소로 이루어진 균등한 요소망이 이용하며 초기 전체문제에 원형 균열은 이산화되지 않는다. 전체 원형 균열을 포함하는 하나의 마스터 국부 영역이 생긴 전체 요소망으로부터 생성되며 495개의 클라우드로 구성되어 있다. 초기 전체문제로부터 공급되는 디리클레(Dirichlet) 경계조건이 각각의 서브 국부문제에 사용되며 정확한 해를 얻기 위해 원형 균열선단 주변으로 요소망이 세분화된다. 마스터 국부 영역은 495개의 서브 국부 영역

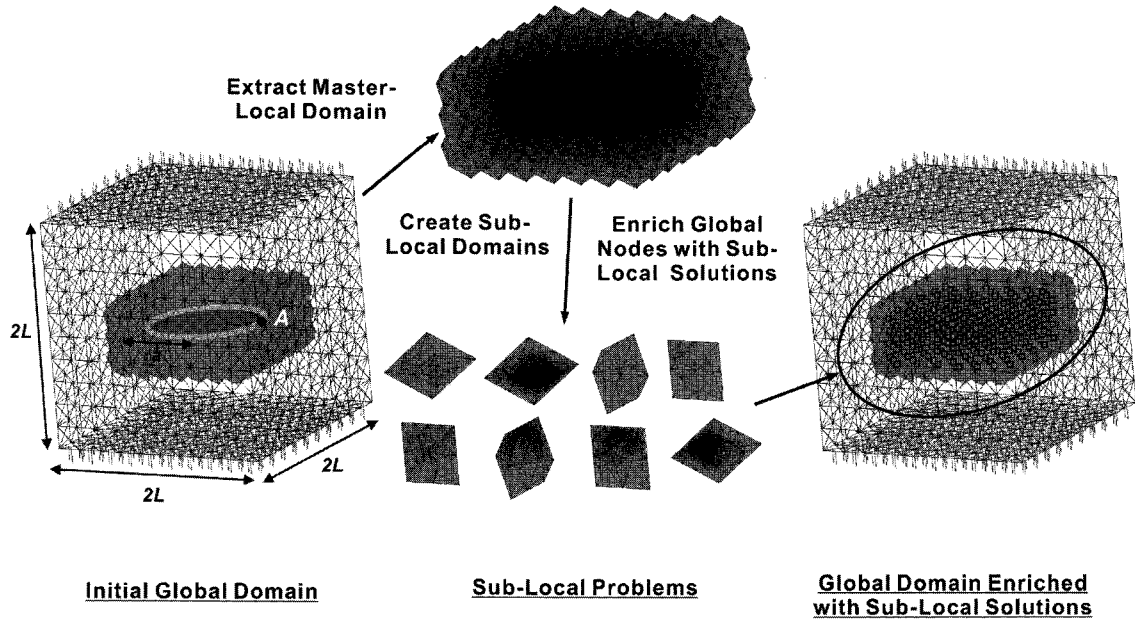


그림 4 정육면체 도메인에 포함된 원형 균열 문제의 해를 병렬연산 일반유한요소 기법을 이용하여 구하는 과정

으로 나누어지며 각각의 서브 국부문제를 병렬연산 기법을 이용해 푼다. 이들 서브 국부문제의 해는 성긴 전체문제에 확장함수로 쓰이게 된다. 전체 그리고 국부문제 모두에 3차의 다항 형상함수가 이용된다.

제안된 기법의 병렬연산 효율성을 평가하기 위해 495개의 병렬연산 해에 필요한 CPU 시간을 공급하는 프로세서의 개수를 바꾸어가며 측정한다. 병렬연산의 속도향상 및 효율성(parallel speed-up and efficiency)은 Quinn(2004)에 소개된 것과 같이 다음의 정의를 이용해 계산된다.

$$\text{병렬연산 속도향상} = \text{직렬연산시 해석시간} / \text{병렬연산시 해석시간}$$

$$\text{병렬연산 효율성} = \text{병렬연산 속도향상} / \text{사용된 프로세서의 개수}$$

그림 5는 3.3장에 소개된 오름 및 내림차순의 분류를 이용하여 495개의 서브 국부문제를 해석하는데 요구된 병렬연산의 속도향상과 효율성을 보여준다. 병렬연산에 이용된 프로세서의 최대개수는 32개이다. 그래프의 수평축은 병렬연산에 이용된 프로세서의 개수를, 좌측과 우측의 수직축은 각각 계산된 효율성과 속도향상을 나타낸다. 내림차순의 분류에 의해 얻어진 속도 향상은 22개의 프로세서를 사용한 경우까지 선형 확장성(linear scalability)을 보여준다. 반면에 오름차순의 분류를 사용한 경우 8개의 프로세서를 사용할 경우부터 선형 확장성에서 벗어나는 결과를 보여주

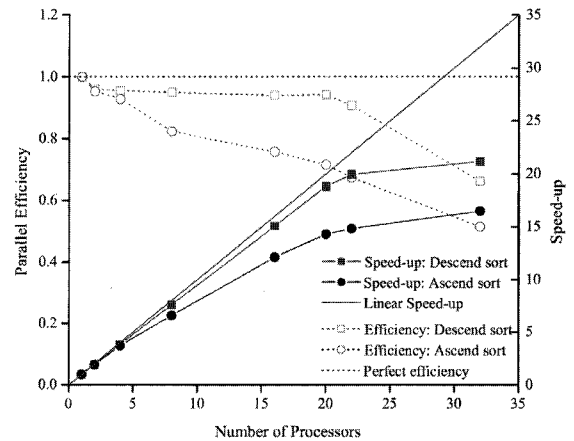


그림 5 원형균열 예제의 병렬연산 효율성

며 전반적인 병렬연산 효율성이 훨씬 낮다. 따라서 3.3장에서 언급된 바와 같이 내림차순으로의 분류를 사용할 경우 병렬연산의 효율성을 높이는 데 큰 도움이 됨을 알 수 있다.

그림 6은 내림차순으로의 분류를 사용하더라도 사용된 프로세서의 개수가 22개가 될 경우 효율성이 갑자기 떨어지는 이유를 보여준다. 20개의 프로세서를 사용한 경우 각각의 프로세서에서 소모된 CPU 시간이 동일하나 22개를 사용한 경우 마지막 2개의 프로세서에서 더 많은 CPU 시간이 소모되었으며, 이들은 오직 하나의 서브 국부문제만이 해석되었음을 알 수 있다. 이는 원형 균열의 선단 주변으로 국부적인 요소망의 세분화를 실시하여 각각의 서브

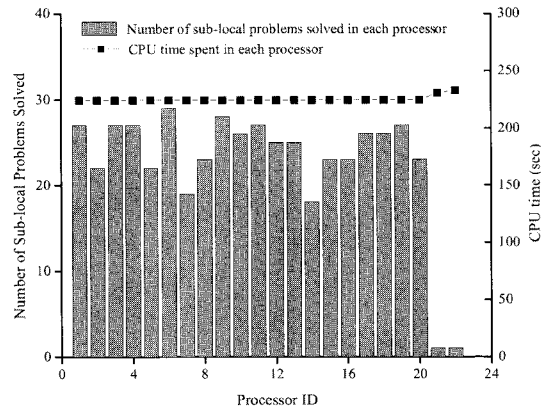
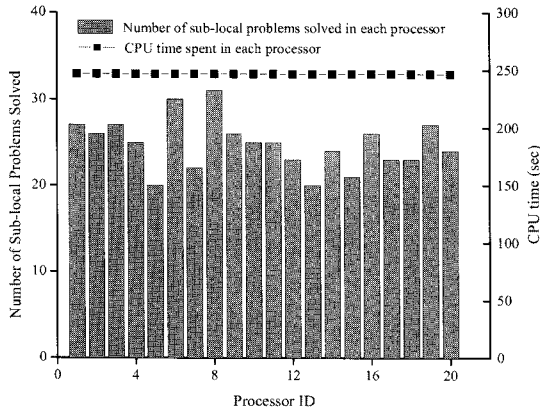


그림 6 원형 균열 예제에서 20개(왼쪽) 및 22개(오른쪽)의 프로세서를 사용한 경우에 대한 각각의 프로세서에의 부하의 균등화 (load balancing). 두 경우 모두 내림 차순으로의 분류가 사용되었다.

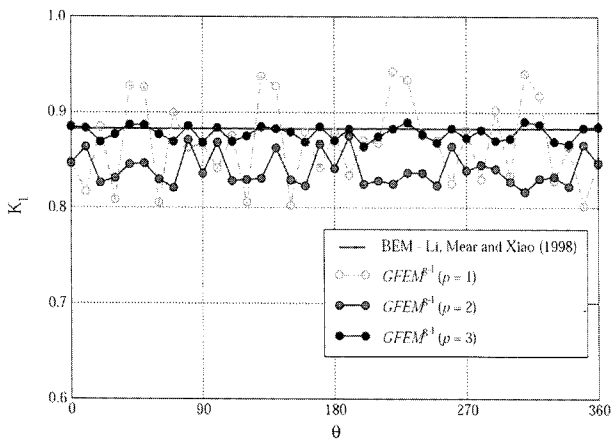


그림 7 원형 균열의 선단을 따라 얻어진 응력강화계수의 분포

국부문제의 크기에 차이가 있고, 마지막 2개의 프로세서에 할당된 2개의 큰 사이즈의 서브 국부문제를 풀기 위해 요구되는 비용(cost)이 나머지 프로세서에서의 평균 계산량 보다 크기 때문에 발생하는 것으로 볼 수 있다.

그림 7은 제안된 병렬연산 GFEMg-1 기법을 이용해 원형 균열 선단을 따라 계산된 첫 번째 모드 응력강화계수(mode I stress intensity factor)의 분포를 보여주며 세 가지 다른 차수의($p = 1, 2, 3$) 다항 형상함수가 이용되었다. 계산된 결과의 정확성을 판단하기 위해 사용된 기준값(reference value)은 Li 등(1998)에서 경계요소법(boundary element method)을 이용해 그림 4의 A점에서 계산된 값이다. 기준 값이 원형균열의 선단의 오직 한 점에서 계산된 값이기는 하지만, 그래프에 나타난 결과는 형상함수의 차수가 증가 될수록 계산된 응력강화계수값이 기준값에 접근하며 진동(oscillation)이 작아짐을 보여준다. 따라서 제안된 기법에 의해 계산된 결과가 정확함을 알 수 있다.

4.2 복잡한 형상을 지닌 구조요소

두 번째 수치 해석 예제는 그림 8에 주어진 복잡한 형상을 지닌 구조요소이다. 이 예제에 쓰인 요소망은 3,849개의 절점 및 15,527개의 요소로 이루어져 있으며, 사용된 탄성 계수 및 포아송 계수는 각각 105 및 0.33이다. 이 예제에서는 많은 자유도를 지닌 문제를 제안된 기법을 이용해 풀 경우 병렬연산의 효율성을 보이는데 초점을 맞춘다.

그림 9는 제안된 일반유한요소 기법을 이용하여 주어진 구조요소를 해석하는 과정을 보여준다. 전체문제의 영역과 동일한 하나의 마스터 국부 영역이 생성되며, 서로 다른 세 레벨(level 1, 2, 3)의 요소망 세분화가 마스터 국부문제에 수행된다. 레벨이 높을수록 요소망의 세분화가 더 많이 이루어진다. 총 3,849개의 서브 국부문제가 요소망이 세분화된 마스터 국부 요소망으로부터 추출되며 이들의 개수는 전체 요소망의 절점 개수와 동일하다. 이들 서브 국부문제는 병렬연산을 이용해 해석되며 이들의 해는 전체문제에 확장함

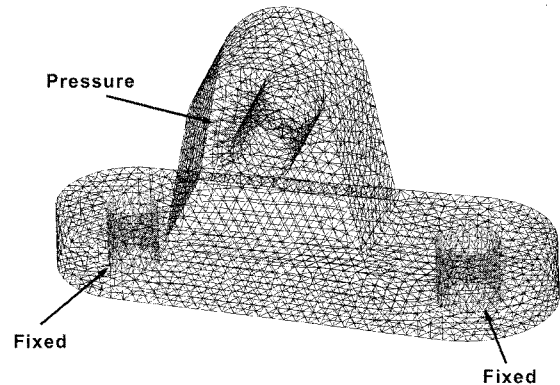


그림 8 복잡한 형상을 지닌 구조요소의 경계 조건 및 요소망

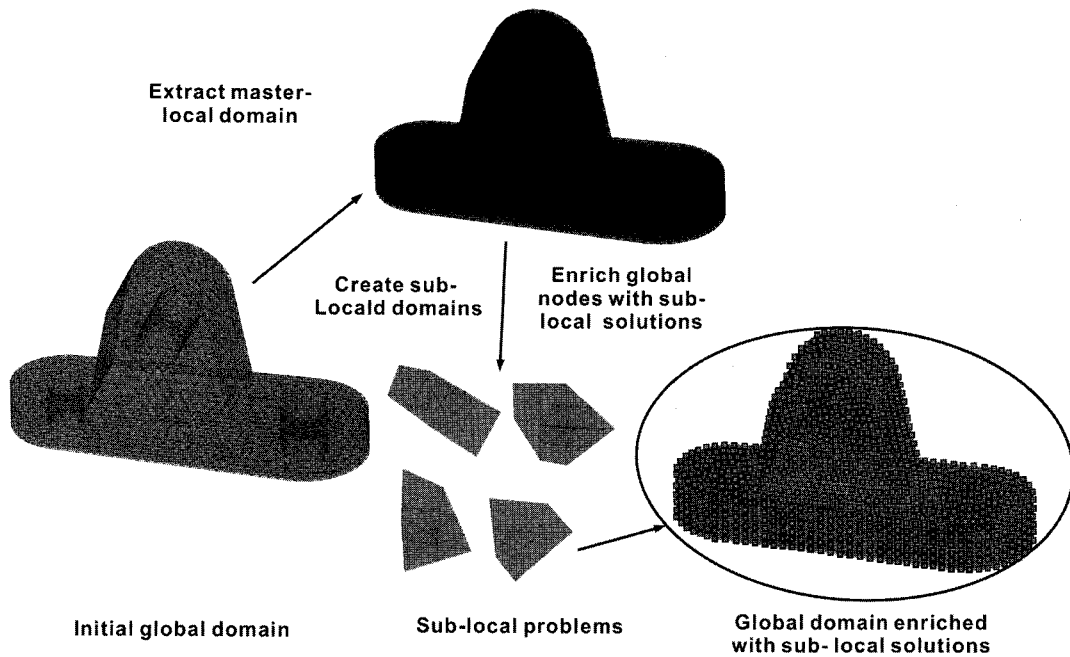


그림 9 복잡한 형상을 지닌 구조요소의 해를 병렬연산 일반유한요소 기법을 이용하여 구하는 과정

표 1 복잡한 형상을 지닌 구조요소의 병렬연산 효율성

프로세서 개수	CPU 시간(sec.)			병렬연산 효율성		
	레벨 1	레벨 2	레벨 3	레벨 1	레벨 2	레벨 3
1	3,534.3			N/A		
2	1,843.6	18,951.8		0.959	N/A	
4	938.0	9,842.6		0.942	0.963	
8	479.4	5,067.5		0.922	0.935	
16	240.6	2,829.5	17,895.8	0.918	0.837	N/A
32	126.0	1,449.1	9,780.7	0.877	0.817	0.913
64		553.8	5,220.6		1.069	0.856
128			2,145.9			1.068

수로 이용된다. 전체 그리고 국부문제 모두에 3차의 다항형상함수가 이용된다.


표 1은 총 128개의 프로세서까지에 대해 3,849개의 서브국부문제를 풀기 위해 요구된 CPU 시간 및 병렬연산 효율성을 보여준다. 요소망의 세분화 레벨이 높아질수록 계산량이 크게 증가하여 프로세서의 최소 개수를 증가시켰으며, 병렬연산의 효율성도 이를 기준으로 계산되었다. 이전 예제와 마찬가지로 모든 개수의 프로세서에 대해 높은 수준의 효율성을 보인다. 경우에 따라 초선형 확장성(superlinear scalability)를 보이기도 하는데, 이는 병렬연산 해석결과를 얻기위해 이용된 슈퍼컴퓨터가 NUMA(non-uniform memory access) 시스템에 기반하여 만들어진 사실과 관련이 있는 듯 하다.

5. 결론

본 원고에서는 병렬연산 기법을 활용한 전체-국부 확장함수를 지닌 일반유한요소법(generalized finite element method with global-local enrichment functions)이 소개되었다. 이 기법은 주어진 문제의 해에 대한 정보가 제한되어 있는 애플리케이션에 대해 확장함수를 수치 해석적으로 생성할 수 있다. 국부문제는 성긴 요소망을 이용해 구해진 전체문제의 해를 국부문제의 경계 조건으로 이용하며 병렬연산 기법을 활용해 해를 구하게 된다. 구해진 국부문제의 해는 다시 확장함수의 형태로 전체문제의 해공간을 확장하는데 사용되며, 이는 적은 자유도를 이용하여 정확한 해를 구하는 것을 가능하게 한다. 병렬연산의 효율성을 높이기 위

해 마스터-서브 국부분제 기법과 서브 국부분제의 내림 차순으로의 분류가 이용되었다. 본 원고에 소개된 수치 해석 예제의 결과는 제안된 기법을 이용해 높은 정확성 및 효율성을 지닌 병렬연산해를 얻을 수 있음을 보여준다.

참고문헌

1. Oden, J.T., Patra, A. and Feng, Y.S. (1992) An hp adaptive strategy. In A. K. Noor, editor, Adaptive, Multilevel and Hierarchical Computational Strategies, ASME, 157 pp. 23-46.
2. Oden, J.T., Duarte, C.A. and Zienkiewicz, O.C.. (1998) A new cloud-based hp finite element method. Computer Methods in Applied Mechanics and Engineering, 153 pp.117~126.
3. Moes, N., Dolbow, J., and Belytschko, T. (1999) A finite element method for crack growth without remeshing International Journal for Numerical Methods in Engineering, 46 pp.131~150.
4. Duarte, C.A., Babuska, I. and Oden, J.T. (2000) Generalized finite element methods for three dimensional structural mechanics problems. Computers and Structures, 77 pp.215~232.
5. Duarte, C.A. and Kim, D.-J. (2008) Analysis and Applications of a Generalized finite element method with global-local enrichment functions Computer Methods in Applied Mechanics and Engineering, 197(6-8) pp.487~504.
6. Kim, D.-J., Pereira, J.P. and Duarte, C.A. (2010) Analysis of three-dimensional fracture mechanics problems: A two-scale approach using coarse generalized FEM meshes. International Journal for Numerical Methods in Engineering, 81(3) pp.335~365.
7. Kim, D.-J., Duarte, C.A. and N.A. Sobh (2011) Parallel simulations of three-dimensional cracks using GFEM. Computational Mechanics, 47 pp.265~282.
8. Quinn, M.J. (2004) Parallel programming in C with MPI and OpenMP. McGraw-Hill, New York.
9. Li, S., Mear, M.E. and Xiao, L. (1998) Symmetric weak-form integral equation method for three-dimensional fracture analysis. Computer Methods in Applied Mechanics and Engineering, 151 pp.435~459. 

[담당 : 유은중, 편집위원]