

중대사고해석 코드(MIDAS)개발을 위한 MP 패키지 개편

박선희*, 김고려, 김동하, 조성원
한국원자력연구소
대전광역시 유성구 덕진동 150

1. 서론

원자력연구소에서는 기존의 중대사고 해석 코드인 MELCOR 를 근간으로 중대사고해석 코드인 MIDAS 를 개발하고 있다. 기존 코드에서의 데이터 저장 방법은 고정적인 크기의 저장장소 관리를 위해 포인터 변수를 사용하고 있어, 코드내부에 대한 이해가 어렵고 모델의 개선 또는 추가의 경우 어려움이 있다.

FORTRAN90 이 발표된 후 저장장소의 동적 할당을 비롯한 장점들이 코드 개발에도 활용되고 있다[1]. 이에 따라 기존 코드의 데이터 저장 및 전달 방법을 재구성하여 코드의 내부를 쉽게 이해할 수 있도록 하였다. 간단한 패키지에 대한 개편 시안을 이미 개발하여 타당성을 분석하였으며[2,3], 전체 코드로 확장하고 있다. 대상 패키지는 물리적인 패키지에 필요한 물성을 모사하는 MP 패키지로 선정하여 코드 구조를 개편하였으며, 개편 이전의 결과와 이후의 결과를 비교하여 검증작업을 수행하였다.

2. 기존 구조

MELCOR code 는 3 부분으로 되어 있으며, 입력자료 점검과 계산에 필요한 file 을 작성하는 MELGEN, 작성된 restart file 을 입력으로 계산을 수행하는 MELCOR, 그리고 PLOT 을 위한 부분이 있다.

MELCOR 코드는 data 를 효율적으로 전달 및 저장하기 위해 4 가지 타입의 저장 장소를 확보하여 사용하고 있으며[4], 각 데이터 타입의 고정된 크기 내에서 사용자의 입력에 따라 저장 용량의 크기를 최적화하여 데이터를 저장한다. 이 과정에서 포인터 변수를 사용하고 있다.

MP 패키지에 대한 포인터 변수가 의미하는 내용은 패키지내의 subroutine 에서 찾아볼 수 있으며, 이를 근거로 MP 패키지에 대한 모듈을 구성하였다. 이것으로 MP 패키지를 개편하였을 때 기존의 MELCOR 와 접목하여 검증이 가능함을 확인하였다.

3. 개발 내용

코드의 전체적인 개편에 앞서 간단한 패키지에 대해 먼저 개편을 하여 타당성을 분석하고 검증 작업을 거친 후 전체 코드로 확장하는 방법으로 진행하였다. MELCOR 1.8.4 를 대상으로 하였으며, 사용 언어는 FORTRAN90 으로 하였다. 데이터 전달 구조는 코드 내부의 이해가 쉽고 사용이 용이하도록 기존 MELCOR 의 단일 배열에서 FORTRAN90 언어의 특성을 활용할 수 있는 도출변수 형태로 바꾸어 패키지별로 모듈화하였다(그림 1).

개편 패키지는 여러 패키지중 물리적인 패키지에 필요한 물성의 모사에 사용되는 MP 패키지로 하였으며, MP 패키지에서 사용하는 변수를 모듈화된 구조로 전환하고, 동적 저장 기법을 이용하여 계산 효율을 높였다. MELCOR 전체의 수행을 제어하는 subroutine 들은 물론, MP 패키지에 속해 있는 70 여개의 subroutine 을 분석하였으며, 개발된 자동 변환 프로그램의 사용으로 수작업을 최소화하여 관련 subroutine 을 변환하였다.

3.1 모듈 구성

Subroutine MPDB1, MPDB2, MPDB3, MPDB6, MPDBE 등에서 subroutine MPDBZ, MPDBY, MPDB4, MPDB7, MPDBF 등을 호출하면서 포인터 변수를 사용하였으며, subroutine MPPS2, MPDB5, MPEDT, MPCHK 등에서는 포인터변수에 대응하는 지역변수 (local variable)를 사용하고 있으며, 이에 근거하여 모듈을 구성하였다.

3.2 Subroutine 개편

Subroutine 의 개편에 앞서 다른 패키지의 개편과 같이 먼저 FORTRAN77 로 쓰여진 기존 MELCOR 코드 전체를 FORTRAN90 언어로 변환하였으며, 비교한 결과도 같음을 확인하였다.

MP 패키지의 subroutine 의 경우, argument 가 전달되는 모든 단계를 확인한 결과, 포인터 변수를 직접 사용하는 subroutine 은 40 여 개였으며, 다른 subroutine 은 포인터 변수를 직접 사용하지 않고 subroutine 내의 local variable 을 사용하여 argument 를 전달하거나 계산하고 있었다. 이러한 subroutine 들 중 포인터 변수와 관련된 변수들에 대해 모듈의 변수들로 변경하였다.

4. 개발결과 및 검증

FORTRAN90 언어로 변환한 MELCOR 수행 결과와 개편된 실행 파일의 MELCOR 의 수행 결과를 MP 패키지와 관련된 주요 변수를 비교하였다. 대부분의 패키지 데이터가 포함된 demo 입력을 작성하여 원자료를 14.950E3 sec 동안 모사하였으며, 그 외 몇개의 경우의 입력에 대해서도 수행되도록 모사하였다.

FORTRAN90 언어로 변환한 MELCOR 수행 결과와 개편된 수행 결과를 COR 패키지와 관련된 주요 변수에 대해 비교하였다. MELGEN 및 MELCOR 의 여러 subroutine 에 대한 개편이 제대로 되었는지의 점검을 위해 data file(restart file)에의 read/write 부분에 대해 확인한 결과, 같음을 확인하였다. 이로써 적어도 data file 에의 read/write 부분에 대한 개편이 제대로 되었음을 확인하였다.

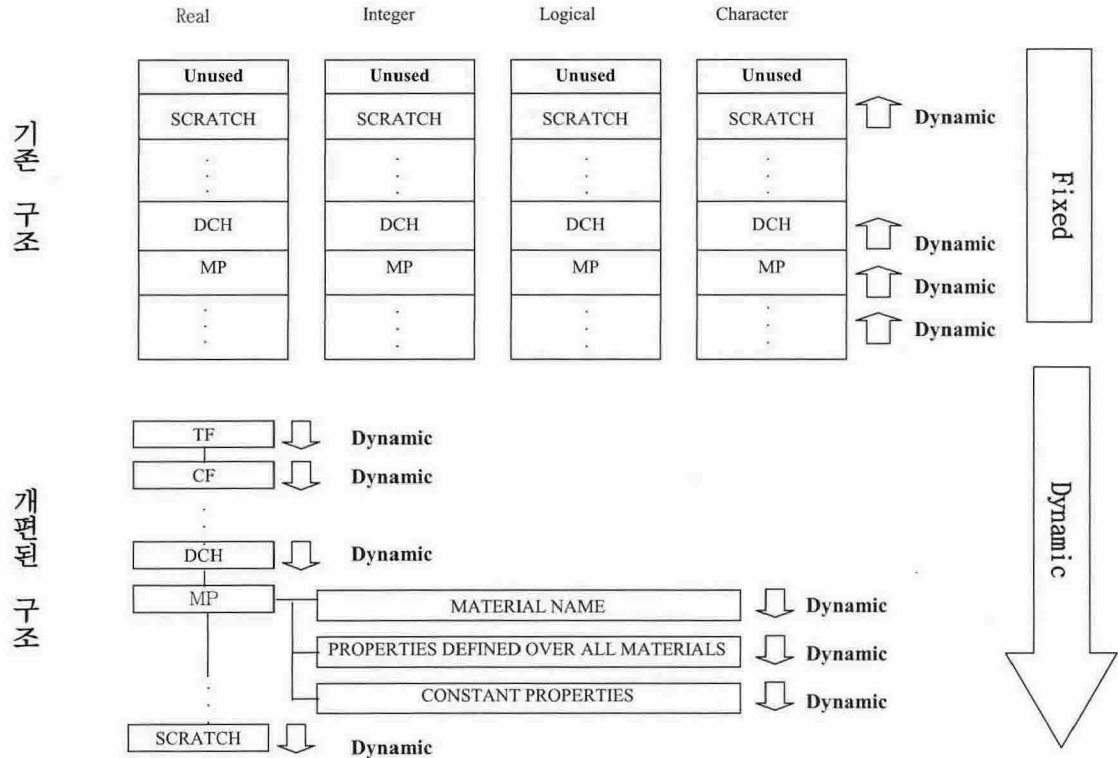


그림 1. 데이터 전달구조 개편의 전체적인 특성

구체적인 비교는 그림 2에 나타내었으며, MP 'CARBON STEEL'과 관련된 열구조의 node 내부 온도를 나타내며, 비교 결과로 보아 MP 팩키지의 변수값이 정확하게 일치하였다. 이 외에 MP 팩키지의 실행에 소요된 CPU time을 비교해 보면 다른 팩키지 개편의 경우 결과와 비슷하게 차이를 보였으며, 이는 부프로그램 호출시 array로의 전환/환원 및 even/odd cycle에 대한 데이터의 달라진 처리로 인해 생길 수 있는 결과로 판단된다.

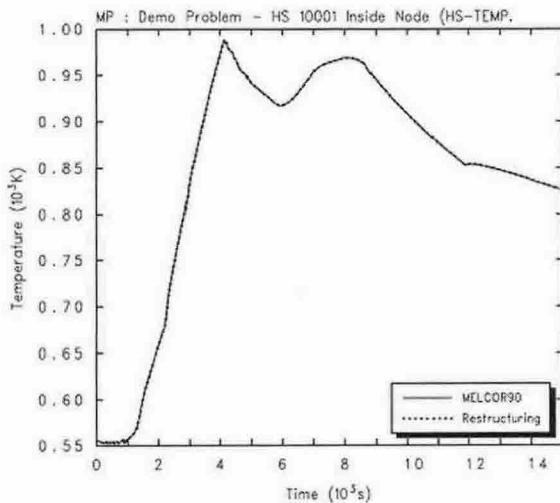


그림 2. MP 'CARBON STEEL'과 관련된 열구조

5. 결론 및 추후 연구 계획

MIDAS에서의 데이터 저장 및 전달방법을 재구성하기 위하여 기존 MELCOR 코드의 전체적인

데이터 흐름을 분석하고, 특히 MP 팩키지에 대한 데이터 정보 및 포인터 변수를 분석하여 MP 팩키지에 적용할 모듈을 구성하였다. 이 과정에서 포인터 변수를 없애고, 대신 직접변수를 사용하도록 데이터구조를 재구성하였다.

새로 구성된 모듈을 사용하여 MP 팩키지의 관련 subroutine을 개편하였으며, MP 팩키지의 개편결과, 코드의 전반에 걸쳐 용이한 변수 파악과 기존 코드의 개선이나 새로운 모델의 접목이 가능한 코드 구조로 확장할 수 있음을 확인하였다.

따라서 본 논문에서 제시된 MP 팩키지의 개편 내용을 동일한 방법에 의해 각 팩키지별로 적용할 수 있도록 지속적으로 확장하여 전체 코드를 변환하고 있으며[5], 우리 실정에 맞도록 코드를 국산화하기 위해서는 기존 모델의 개선에 대한 연구와 새로운 모델의 추가에 대한 연구가 지속적으로 요구된다.

REFERENCES

[1] A Multi-Dimensional Thermal-Hydraulic System Analysis Code, MARS 1.3.1, Vol.31, Number 3, pp.344-363, June 1998.
 [2] 박선희, 송용만, 김동하, A Restructuring Proposal for MIDAS, 한국원자력학회 춘계학술발표회 논문집, 2000.
 [3] S.H.Park, H.D.Kim, D.H.Kim, Y.M.Song, B.D.Chung, Development of Restructuring Template for MELCOR, 5th PSAM Conference, Japan, 2000.11.26 - 2000.12.2
 [4] 김동하, 박선희, MELCOR 코드 구조 분석, KAERI/TR-1543/00, June, 2000.
 [5] 김동하, 외 12인, 중대사고 위해도 완화전략분석 종합전산코드 개발, KAERI/RR-2216/2001, pp 302-321, May, 2002