

ASME B&PV Code Section III NB-3200의 규정에 따른 응력해석 결과 후처리 통합 Program

남궁 인, 김 인용, 조 충희
한국원자력연구소

요 약

ASME B&PV Code Section III NB-3200의 규정은 원자로 관련 1등급 부품의 설계시 지켜야 할 사항이다. 이 규정은 운전조건별로 허용응력에 대한 분류를 하여 허용한도를 규정하고 있다. 따라서 응력해석시 이 규정을 적용하기 위해 해석결과의 검색, 추출정리, 추가계산등 응력해석 후속작업을 위한 통합 program을 awk 언어를 사용하여 개발하였다. 이 통합 program은 ASME에 규정된 응력별로 여러개의 awk program module로 작성하였고 각각의 모듈을 통합하는 UNIX script file로 구성되어 있다. 각각의 모듈은 독립된 batch 작업이 가능하고, 이것을 모두 연계한 batch 작업 역시 가능하도록 하였다. 문서작성시 도표작성을 용이하게 하기 위해 후처리 결과가 하나의 디렉토리에 저장되도록 하였다.

1. 서 론

원자로관련부품의 유한요소 응력해석시 범용 응력해석코드인 ANSYS를 사용하여 해석하는 경우 postprocessor에 선형화 응력모듈과 피로해석모듈이 포함되어있어서 ASME에 규정된 응력유형의 분류 및 피로해석을 지원하고 있다. 그러나 운전조건별 적용에 대해서는 해석자가 직접해야 한다. 그래서 해석대상부품이 복잡해지고 커질수록, 응력해석결과 분석에 많은 시간이 필요하다. 그리고 유한요소 응력해석 결과는 그분량이 방대하기 때문에 정리하여 필요한 부분만을 요약하여 해석보고서에 삽입해야 할 필요성이 있다. 이렇게 필요한 부분만을 추려서 첨부하는 작업외에 응력의 형태별 분류 통합 및 추가계산을 하여 규정된 응력한도치와 비교하는 작업이 요구된다.

여기서는 ANSYS를 사용하여 해석한 결과에 대한 후처리작업 프로그램을 Apollo UNIX 환경에서 개발하였다. 이를 위해 awk언어를 사용하였고, 표1과 표2에 분류된 응력을 각각의 모듈로 만들었다. 각각의 모듈은 따로 batch작업이 가능하도록하였고, 전체를 모두 통합하는 shell script를 이용하여 응력해석 전과정을 하나로 통합하여 수행할 수 있도록 하였다. 이렇게 이단계로 설계함으로써, 응력해석 수행없이 postprocess중 일부분만의 작업변경시 해당모듈만 사용하면 되도록 하였고, 응력해석과정의 경계조건 및 모델자체의 변경사항 발생시에는 전체통합 shell script를 사용하여 하나의 batch job으로 재해석 및 후처리를 할 수 있도록 하였다.

awk언어를 이용한 이유는 awk언어가 단시간에 프로그램을 개발할 수 있고, 패턴찾기 기능이 뛰

어나고, UNIX와의 연계가 편리하기 때문이다. awk언어는 이외에 연산기능과 행열표현도 가능한 언어로서, C언어와 UNIX의 shell program의 중간쯤되는 interpreter 언어이다. awk언어의 사용은 program 개발시간을 단축할수있고, 손쉽게 program을 변경할수 있으며, compile작업이 필요없고, 필요시 C언어로 변환하여 효율을 높일수있는 장점등이 있다.

2. ASME B&PV Code Section III Subsection NB-3200의 응력한도 규정

ASME B&PV Code Section III Subsection NB는 원자로관련 1등급부품의 설계규정이다. 이중 NB-3200은 응력해석에의한 설계시 지켜야할 규정이다. 이 NB-3200규정은 탄성해석과 소성해석으로 구분할수있고, 소성해석은 탄성해석으로 응력규정 한도를 넘길경우 선별적으로 적용할수있는 규정이다. 이규정은 응력해석 결과에대한 응력의 분류및 부품의 재질에따른 허용응력 한도치로 이루어져있다. NB-3200의 규정을 적용하여 상세응력해석을 할경우에는 ANSYS나 ABAQUS와 같은 유한요소 응력해석 코드를 이용하게된다.

표1은 볼트를 제외한 부품의 응력한도에대한 규정을 나타내고, 표2는 볼트에대한 응력한도 규정을 나타낸다. 표1 과 표2에 나타낸 규정은 일반적인경우 따라야할 전반적인 규정이다. ASME의 규정에대한 배경및 정신은 참고문헌 4,5,6 에 설명되어있다. ASME의 규정은 응력을 분류하여 그 한도치를 각 운전조건별로 규정하였다. 이를위해 ANSYS의 경우 postprocessor에 다음은 후처리를 위한 각각의 awk program module의 pattern 찾기부분에대해 설명한다. 선형화응력모듈과 ASME의 피로해석 규정에 입각한 피로해석모듈이 포함되어 있다. 이외에 특별응력 한도치로서, 지지응력(bearing stress), 순수전단 (pure shear) 와 삼축응력 (triaxial stress) 등이 있고, 재료의 취성파괴방지규정 (열응력과 압력 및 기계적하중등 기타하중에의한 응력 계산이 필요) 등이 있다. 볼트의 경우 설계조건규정은 응력해석이 필요치않은 볼트단면넓이의 최소치에대한 규정이다. 따라서 이러한 모든 작업을위해 각각의 결과에대한 추가적인 작업이 필요하다.

3. 모듈 설명

위에 설명한바와 같이 유한요소 응력해석의 결과를 문서로 작성하기위해서 많은 추가적인 작업이 필요하다. 그림1은 각모듈의 기본이되는 flow diagram이고, 그림 2는 응력해석시 구성하게되는 전체 디렉토리구조이다. 그림 2의 /component에 딸린 디렉토리는 각 기능별로 해석을 수행하기 위해 분리되어있다. 이중 /postprocess의 종속 디렉토리는 표1에 보인 바와 같이 응력성분별로 구분하였고, 최종결과가 /document에 복사되어 모아지도록 된다. 후처리를 위한 awk program들은 /awk_prog에 모여져서 통합 관리되고, ANSYS run 및 후처리작업을 수행하도록 UNIX shell script가 각 디렉토리에 작성되어 있다 (표 3 참조). 표 4에 후처리를 위한 모듈의 이름과 기능에 대한 간략한 설명이 있다. 이러한 awk program과 shell script는 Apollo UNIX 환경하에서 개발되었고, DOS용 awk interpreter를 사용하여 PC에서도 별도의 작업없이 그대로 활용이 가능하다.

표 5에 UNIX shell script중 post_all과 lineariz가 나와있고 다른 shell script도 이와 유사하다. Shell scrip post_all의 경우 각 디렉토리로 바꾸는 작업과 바뀐 디렉토리를 나타내는 작업 그리고 해당하는 작업등이 반복되어 진다. 그리고 shell script lineariz의 경우 ANSYS의

postpross run을 마치고 그 결과 각 운전조건별로 생성되는 파일에 대해 AWK 프로그램으로 작업을 한다. lineariz는 선형화응력과 운전조건 A 및 B에 대한 3Sm응력계산 결과도 검색 취합한다. 마지막으로 이렇게 생성된 파일들을 /documents 디렉토리로 복사 저장한다. 그리고 ANSYS 입력파일이 다른 ANSYS run결과를 이용하는 경우, 예를 들면 ANSYS post run이 static run의결과를 사용하기 위해 /static 디렉토리의 결과파일 이용, 그에 맞는 GET 명령어가 입력파일서두에 삽입되어 있어야한다.

표 6에 awk 프로그램 모듈의 하나인 lineariz.awk를 나타내었는데, ANSYS의 선형화응력 결과는 선형화응력계별별로 프린트된다. 그래서 처음은 "**** POST1 LINEARIZED"로 시작한다. 다음으로 "** BENDING **"으로부터 굽힘응력을 출력하고, "** MEMBRANE PLUS BENDING" 다음에 막응력+굽힘응력, "** PEAK **"다음에 최대응력, 그리고 "** TOTAL **"다음에 전체응력의 순으로 출력된다. 이러한 패턴이 각 선형화응력계별 별로 출력된다. 그래서 각 선형화 계별별 응력을 저장한후 전단계의 응력과 비교선택하여 최대치를 찾아서 정리 출력한다. 마지막 라인에 있는 \$1 > \$2의 \$1은 입력화일을 \$2는 출력파일을 나타내는 변수이다.

4. 결론

유한요소해석후 ASME code를 적용하는데 따른 해석 후처리작업의 능률을 올리기 위하여 awk언어를 이용하여 후처리 프로그램을 개발하였다. ASME의 규정에 따라 응력 성분별로 분류하여 각모듈을 개발하였고, 모듈사이를 UNIX shell script로 조합 운영하는 방식을 택하였다. 해석결과의 후처리를 수작업하는데 따르는 오류방지 및 시간단축 입력 모델이나 경계조건이 바뀌었을 경우 신속한 해석결과를 얻을 수 있다. 후처리의 최종결과가 하나의 디렉토리에 모아지게 하여, 문서작성시 도표 작성이 용이하도록 하였다. 후처리 프로그램의 패턴검색부분을 수정하여 ANSYS가 아닌 다른 유한요소해석코드를 사용한 경우에도 적용할 수 있다. 그리고 이 프로그램은 Apollo UNIX 환경아래서 개발되었지만, DOS용 awk 언어를 사용하여 PC 환경에도 사용이 가능하다.

5. 참고 문헌

- 1) ASME Boiler and Pressure Vessel Code, 1995 Edition, ASME
- 2) Alfred V.Aho, Brian W.Kernighan, Peter J. Weinberger, The AWK Programming Language, 1988, Addison-Wesley
- 3) Brian Kernighan, Rob Pike, The UNIX Programing Environment, 1984, Prentice-Hall
- 4) W.C.Kroenke, "Classification of Finite Element Stresses According to ASME Section III Stress Categories", Pressure Vessels and Piping, Analysis and Computers, ASME, June 1974.
- 5) ASME, "Criteria of the ASME Boiler and Pressure Vessel Code for Design by Analysis in Sections III and VIII, Division 2", ASME, 1968.
- 6) ASME, "Refernce Fracture Toughness Procedures Applied to Pressure Vessel Materials", MPC-24, ASME, Dec. 1984.

표 1. 1등급부품(볼트제외)의 응력 허용치
(Class 1 Stress Limits for Other Than Bolts, Elastic Analysis)

응력 유형		설계 조건	운전 조건				시험 조건
			Level A	Level B	Level C	Level D	
일차응력 허용치	P_m	S_m	-	$1.1S_m^{(1)}$	Larger of $1.2S_m$ or S_y	Section III, Appendix F	$0.9 S_y$
	P_L	$1.5 S_m$	-	$1.65S_m^{(1)}$	Larger of $1.8S_m$ or $1.5S_y$	Section III, Appendix F	$1.35S_y$ for $P_m \leq 0.67S_y$ $2.15S_y - 1.2P_m$ for $0.67S_y < P_m \leq 0.90S_y$
	$P_L + P_b$						
반복응력허용치 (일차+이차응력)	$P_L + P_b + Q$	-	$3 S_m$	$3 S_m$	-	-	-
피로수명(일차+이차+최대응력)	$P_L + P_b + Q + F$	-	누계사용계수 $\leq 1.0^{(2)}$ (cumulative usage factor ≤ 1)			-	-
특별응력 허용치	지지응력 (bearing str)	S_y	S_y	S_y	S_y	-	S_y
	순수전단 (Pure shear)	$0.6 S_m$	$0.6 S_m$	$0.6 S_m$	$0.6 S_m$	-	$0.6 S_m$
	삼축응력 ($\sigma_1 + \sigma_2 + \sigma_3$)	$4 S_m$	$4 S_m$	$4 S_m$	$4 S_m$	-	$4 S_m$
취성 파괴 방지		-	Sect. III, Appendix G-2200 : $K_I < K_{IR}$	Sect. III G-2300	-	-	Test Temp. $\geq RT_{NDT} + 60^\circ F$

주:

- (1) 운전조건 B (Level B)의 압력이 설계조건 압력보다 클 경우 적용한다.
 (2) 운전조건 C (Level C)의 하중주기를 피로응력 계산에 포함해야한다는 명확한 규정은 없으나, 보수적인 피로수명 계산을 위해 포함시킨다. 응력집중계수를 최대한 5이하로 적용한다.

표 2. 1등급 볼트의 응력 허용치 (Class 1 Stress Limits for Bolts)

응력 유형	설계조건	운전 조건				시험 조건
		Level A	Level B	Level C	Level D	
볼트 단면적	$A_b > A_m$ (App. E)	-	-	-	-	-
단면 평균 응력 강성도	-	$2 S_m$	$2 S_m$	$2 S_m$	App. F	$2 S_m$ assumed ⁽¹⁾
단면 최대응력 강성도	-	$3 S_m$	$3 S_m$	$3 S_m$	App. F	$3 S_m$ assumed ⁽¹⁾
피로수명	-	누계사용계수 $\leq 1.0^{(2)}$		-	-	-
취성 파괴 방지	-	App. G-2200 : $K_I < K_{IR}$		Appendix G-2300	-	Test Temp. $\geq RT_{NDT} + 60^\circ F$

주:

- (1) ASME 코드에 시험조건에 대한 별도의 규정이 없으나, 보수적 계산을 위해 운전조건과 같은 규정 적용.
 (2) 응력집중계수 최소한 4 이상 적용.

P_m : 막응력 P_L : 극부막응력 P_b : 굽힘응력 Q : 이차응력 F : 최대응력
 A_b : 실제 볼트 단면적 A_m : 최소 볼트 단면적 요건
 S_m : 응력 강성도 (Appendix I참조) S_y : 항복 응력 (Appendix I참조)
 K_I : 응력강성계수 RT_{NDT} : 취성 온도 K_{IR} : 기준임계응력강성계수

표 3. UNIX shell scripts

이름	디렉토리	기능
run_all	/component	ANSYS run 및 Post run 총합
run_temp	/temp	ANSYS temperature run
run_static	/static	ANSYS static run
post_all	/postprocess	Postprocess run 총합
lineariz	/lineariz	Linearized str., 3Sm str. run
fatigue	/fatigue	Fatigue analysis post run
triaxial	/triaxil	Triaxial stress post run
bearing	/bearing	Bearing stress post run
non_duct	/non_ductile	Non-ductile stress post run
bolt_nod	/bolt_str	Max. bolt nodal stress post run
plot_pscr	/plot	Postscript plot run
plot_intl	/plot	Interleaf plot run

표 4. AWK Program Modules

이름	기능
strip.awk	Output 으로부터 불필요한 comments 제거
lineariz.awk	Max. linearized S.I. 선별 정리
fatigue.awk	Max 3Sm, fatigue usage factor 선별 정리
bearing.awk	bearing stress 계산 정리
triaxial.awk	Max. 삼축응력 선별 및 계산
bolt_nod.awk	Max. bolt nodal stress 선별 정리
fig_pscr.awk	Postscript 그림축소 및 그림번호 재부여
fig_intl.awk	Interleaf 그림축소 및 그림번호 재부여

표 5. UNIX shell script 예

File : post_all	File : lineariz (3Sm 응력 계산 포함)
cd ~/comp1/post/lineariz	/ansys/ansys.e < lineariz.ans > lineariz.out
ehco ~/comp1/post/lineariz	/ansys/ansys.e < A_3sm.ans > A_3sm.out
lineariz	/ansys/ansys.e < B_3sm.ans > B_3sm.out
cd ~/comp1/post/fatigue	../awk_prog/strip.awk B_linear.out B_linear.txt
ehco ~/comp1/post/fatigue	../awk_prog/strip.awk C_linear.out C_linear.txt
fatigue	../awk_prog/strip.awk DG_linear.out DG_linear.txt
cd ~/comp1/post/bearing	../awk_prog/strip.awk T_linear.out T_linear.txt
ehco ~/comp1/post/bearing	../awk_prog/strip.awk T_linear.out T_linear.txt
bearing	../awk_prog/lineariz.awk B_linear.txt B_linear.max
cd ~/comp1/post/triaxial	../awk_prog/lineariz.awk C_linear.txt C_linear.max
ehco ~/comp1/post/triaxial	../awk_prog/lineariz.awk DG_linear.txt DG_linear.max
triaxial	../awk_prog/lineariz.awk T_linear.txt T_linear.max
cd ~/comp1/post/non_duct	../awk_prog/fatigue.awk A_3sm.txt A_3sm.max
ehco ~/comp1/post/non_duct	../awk_prog/fatigue.awk B_3sm.txt B_3sm.max
non_duct	
cd ~/comp1/post/bolt_str	cp -s *.ans ../documents/
ehco ~/comp1/post/bolt_str	cp -s *.txt ../documents/
bolt_str	cp -s *.max ../documents/
cd ~/comp1/post/plot	rm *.out
ehco ~/comp1/post/plot	
plot_pscr	
plot_intl	

표 6. awk program module 예

```

# File : lineariz.awk
# An awk program to extract Stress Intensity value from ANSYS
# Linearized Stress calculation. This program extracts Bending SI,
# Membrane + Bending SI, Peak SI and Total SI.
awk '
{
    printf("==== Stress Intensity Categories =====\n")
    printf("BENDING SI MEM+BENDING SI PEAK SI TOTAL SI")
    printf(" LS# LOC NODE #\n")
    while(getline > 0) { # Read File line by line & Search for Pattern
        if($1 == "*****" && $2 ~ /POST1/ && $3 ~ /LINEARIZED/)
            getline; i_node = $4; o_node = $8
            getline; getline; ls = $3
        }
        if($1 == "***" && $2 ~ /BENDING/ && $3 == "***") {
            store_str(b); i=1 # store Bend Stress
        }
        if($1 == "***" && $2 ~ /MEMBRANE/ && $3 ~ /PLUS/ && $4 ~ /BENDING/)
        { store_str(mb); j = 1 # Store membrane + bending str.
            if(max_str[0] < mb[2]) {
                max_str[0] = mb[2];
                node[0] = "in_node " i_node " out_node " o_node;
            }
        }
        if($1 == "***" && $2 ~ /PEAK/ && $3 == "***") {
            store_str(p); k=1 # Store peak stress.
        }
        if($1 == "***" && $2 ~ /TOTAL/ && $3 == "***") {
            store_str(t); l=1 # Store total stress.
        }
        if(i && j && k && l) { # Print stored stress.
            printf("%13.2f%13.2f%13.2f%13.2f",b[1],mb[1],p[1],t[1])
            printf(" :%3d, I %5d %4d\n",ls,i_node, o_node)
            printf("%13.2f%13.2f%13.2f%13.2f",b[2],mb[2],p[2],t[2])
            printf(" :%3d, C %5d %4d\n",ls,i_node, o_node)
            printf("%13.2f%13.2f%13.2f%13.2f",b[3],mb[3],p[3],t[3])
            printf(" :%3d, O %5d %4d\n",ls,i_node, o_node)
            store_max(node, max_str) # Compare & store max. str.
            i = j = k = l = 0;
        }
    }
}
'
function store_str(str) { # Function to store S.I. only
    getline; getline; getline; getline;
    getline; str[1] = $5
    getline; str[2] = $5
    getline; str[3] = $5
}
# Store maximum stress.
function store_max(nod, str) {
    compare(nod, str, b, 1)
    compare(nod, str, mb, 2)
    compare(nod, str, p, 3)
    compare(nod, str, t, 4)
}
# Function to compare and store max. str.
function compare(nod, mx_str, new_str, i) {
    if(mx_str[i] < new_str[1]) {
        mx_str[i] = new_str[1]
        nod[i] = "At inside node " i_node
    }
    if(mx_str[i] < new_str[2]) {
        mx_str[i] = new_str[2]
        nod[i] = "At center of" i_node "& " o_node
    }
    if(mx_str[i] < new_str[3]) {
        mx_str[i] = new_str[3]
        nod[i] = "At outside node " o_node
    }
}
END { # Print maximum S.I. and node number.
    printf("\n\n STRESS SUMMARY \n\n")
    printf("Stress Category Max. Stress Node Location\n")
    printf("Mem. Str. %12.3f psi %s\n", max_str[0], node[0])
    printf("Bend Str. %12.3f psi %s\n", max_str[1], node[1])
    printf("M+B Str. %12.3f psi %s\n", max_str[2], node[2])
    printf("Peak Stress %12.3f psi %s\n", max_str[3], node[3])
    printf("Total Stress %12.3f psi %s\n", max_str[4], node[4])
}
$1 > $2

```

그림 1. 각 모듈 flow diagram

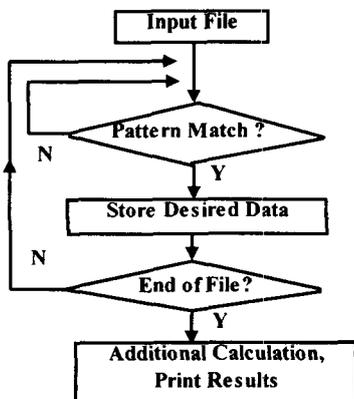


그림 2. 디렉토리 구조

