한국전산구조공학회
1994년도 가을학술발표회 논문집

철근콘크리트 쉘 요소의 강성행렬 계산을 위한 벡터알고리즘
Vector Algorithm for RC Shell Element Stiffness Matrix

민창식* A. K. Gupta**
Min, Chang-Shik Ajaya Kumar Gupta

ABSTRACT
A vector algorithm for calculating the stiffness matrices of reinforced concrete shell elements is presented. The algorithm is based on establishing vector lengths equal to the number of elements. The computational efficiency of the proposed algorithm is assessed on a Cray Y-MP supercomputer. It is shown that the vector algorithm achieves scalar-to-vector speedup of 1.7 to 7.6 on three inelastic problems.

1. 서 론

철근콘크리트 쉘 구조물의 현행 설계방법은[1] 외력에 의해서 야기되는 응력은 탄성해석에 의해서 구하고, 이 응력에 대한 보강철근의 설계는 특정부분의 극한거동에[3,12] 근거를 두어 행하고 있다. 이러한 설계방법이 라아멘과 같은 철근콘크리트 구조물에 대해서는 현재 일반적으로 사용되고 있는 방법이며, 이 설계방법의 타당성 설명은 탄성해석에 의해서 구한 응력이 평형해라는 사실과 이론적으로 소성이론의 하계정리(Lower bound theorem)에서 찾을 수 있을 것이다. 하계정리에 의하면 평형상태로 부터 구한 응력을 사용하여 설계한 구조물의 설계하중은 구조물의 실제의 극한강도에 비해서 언제나 하계에 있다는 정리이다. 그러나 철근콘크리트 재료는 하계정리가 적용되기 위한 조건인 탄성-완전소성 재료가 아니기 때문에 엄밀히 말한다면 적용할 수 없을 것이다. 그러나 다른 종류의 철근콘크리트 구조물에 대해서는 현행 설계방법의 타당성이 그 동안의 수많은 시험과 해석, 그리고 경험에 의하여 확립 되었다고 볼 수 있다. 그러나 철근콘크리트 쉘 구조물에 대해서는 경험과 실험의 미비로 인하여 아직까지 현행의 설계개념에 대한 확실한 기반을 구축하지 못하고 있는 형편이다. 쉘의 두께가 부재의 다른 치수에 비해서 매우 얇기 때문에 실험을 수행하기 매우 어려울 뿐만 아니라 비용과 시간이 많이 들기 때문에, 근래에는 컴퓨터를 이용한 유한요소기법으로 쉘의 극한거동 연구를 해오고 있다[3,13,15,16,18,19]. 그러나, 종래의 IBM 3081과 같은 스칼라 컴퓨터로 극한거동 해석을 하게 되면 유한요소 수가 작은 문제를 해석하기 위해서도 과다한 CPU(Central Processing Unit) 시간이 소요되기 때문에, 더 많은 그리고 더 자세한 해석을 하고자 하는 시도에 크나 큰 제한으로 작용하여 왔었다[3].

근래에 와서 강력한 슈퍼컴퓨터의 출현은 몇년전 만하더라도 컴퓨터의 용량제한 때문에 풀지 못했던 문제들을 이제는 상당한 범위까지 풀 수 있다는 가능성이 확대 되고 있다. 단순히 종래에 스칼라 컴퓨터에서 개발된 프로그램을 이 벡터 슈퍼컴퓨터에 적용하여 사용하게 되면 제한된 효과 밖에는 얻지 못하게 된다. 따라서 최대의 효율을 얻기 위해서는 이 벡터컴퓨터의 하드

* 제주시 제주대학교 해양·토목공학과 690-756. 전임강사.
** Center for Nuclear Power Plant Structures, Equipment and Piping. North Carolina State Univ., Raleigh, NC 27695-7908. 토목공학과 교수 및 소장.

웨어와 컴파일러에 맞도록 프로그램을 다시 짜거나, 또는 새로운 알고리즘을 개발해야 된다는 문제가 야기되고 있다[6,7,9,14,17,18,20,21,22,24].

유한요소 프로그램은 크게 나누면 4 모듈로 구분할 수 있다. 즉, (1) 요소 강성행렬 계산, (2) 전체 강성행렬의 조합, (3) 선형 연립방정식의 해, 그리고 (4) 요소의 변형률 및 응력의 계산이다. 상대적으로 간단한 2차원 유한요소 해석 모델에 대한 벡터 알고리즘을 개발하여 IBM 3090-600E VF 슈퍼컴퓨터에 적용한 결과에 의하면[17], 유한요소 프로그램의 4 모듈 중에서 선형 연립방정식의 해를 계산하는 데 가장 많은 CPU 시간이 소모되었다. 따라서 이 모듈을 벡터화 하여 슈퍼컴퓨터에서 최적화 하면 가장 뚜렷한 CPU 시간을 줄이는 효과를 얻을 수가 있으며, 이러한 노력은 광범위하게 시도되고 있는 형편이다[6,8,14,22]. 그러나 나머지의 모듈도 최적화 하게 되면 스칼라계산에 대해서 벡터계산을 할 때 추가적으로 10-40% 의 속도증가의 효과를 얻을 수 있을 뿐만 아니라, 비탄성해석에서는 반복계산 과정에서 나머지 모듈의 계산도 매 반복마다 시도해야 하므로, 최적상태의 알고리즘을 개발하기 위해서는 나머지 모듈도 최적화 하여야 할 것이다[17].

원래에 Akbar와 Gupta[3]에 의해서 IBM 3081 스칼라 컴퓨터에 개발한 철근콘크리트 쉘의 비탄성해석 유한요소 컴퓨터 프로그램을 벡터화 하여 North Carolina Supercomputing Center (NCSC)에 있는 Cray Y-MP 슈퍼컴퓨터에 적용하였다. 본 연구에서는 균열이 나타나기 전과 난 뒤의 철근콘크리트 쉘 요소 강성행렬 계산의 벡터 알고리즘을 제시하여 Cray Y-MP 나 IBM ES/9000 등과 같은 벡터 슈퍼컴퓨터가 갖는 이점을 최대 한도로 이용할 수 있도록 시도하였다. 유한요소로는 4-절점 고매개변수가 사용되었고[2], 휨응력이 콘크리트 요소의 균열이나 보강철근의 항복에 영향을 미치지 않는다는 Akbar와 Gupta가[3] 한 가정을 그대로 사용하였다. 이 가정은 많은 쉘 구조물의 거동이 평면-막응력에 의해서 좌우되고 Akbar와 Gupta가 개발한 프로그램을 이러한 쉘 구조물을 해석하는데 사용하므로 써 정당화할 수 있을 것이다.

2. 요소 강성행렬과 수치적분

강성행렬 계산은 Gupta와 Mohraz[10]가 제안한 방법과 유사한 알고리즘을 사용하였다. 강성행렬 $[k]$는 크기가 6x6 인 nn x nn 개의 작은 행렬로 나눌 수 있다. 여기에서 nn은 한 요소의 절점 수이며, 여기에서는 4 이다. 절점 i 의 힘과 절점 j 의 변위에 관계시키는 작은 행렬 $[k_{ij}]$는 평면강성 $[k_{p_{ij}}]$와 횡방향 전단강성 $[k_{s_{ij}}]$로 분류할 수 있다.

구성특성이 한 요소 안에서 불변 한다면, 작은행렬 $[k_{p_{ij}}]$는 아래와 같이 표현할 수 있다.

$$[k_{p_{ij}}]_{6x6} = h\begin{bmatrix} [P] & 0 \\ 0 & \frac{1}{3}[P] \end{bmatrix}, \quad [P]_{3x3} = \begin{bmatrix} (E_1+E_s\rho_{\bar{x}})[p_1]+E_2[p_2]+E_3[p_3] \\ +(E_4+E_s\rho_{\bar{y}})[p_4]+E_5[p_5]+E_6[p_6] \end{bmatrix}, \qquad (1)$$

여기에서 h는 요소의 두께이고, 요소 내에서 불변 한다고 보았다. $E_1, E_2, E_3, E_4, E_5, E_6$는 콘크리트 평면응력의 구성행렬 요소들이고 아래의 식에 의해서 정의 된다.

$$\begin{pmatrix} \sigma_{\bar{x}} \\ \sigma_{\bar{y}} \\ \tau_{\bar{x}\bar{y}} \end{pmatrix} = \begin{bmatrix} E_1 & E_2 & E_3 \\ & E_4 & E_5 \\ sym & & E_6 \end{bmatrix} \begin{pmatrix} \varepsilon_{\bar{x}} \\ \varepsilon_{\bar{y}} \\ \gamma_{\bar{x}\bar{y}} \end{pmatrix}, \qquad (2)$$

여기에서 $[\sigma_{\bar{x}}\ \sigma_{\bar{y}}\ \tau_{\bar{x}\bar{y}}]^T$와 $[\varepsilon_{\bar{x}}\ \varepsilon_{\bar{y}}\ \gamma_{\bar{x}\bar{y}}]^T$는 쉘 요소의 직교 좌표계, $\bar{x}$와 $\bar{y}$에 대한 평면응력과 변형률이다. 균열전의 콘크리트 요소에서 E_3와 E_5는 0이며, 보강철근에서 $\sigma_{\bar{x}} = E_S\rho_{\bar{x}}\varepsilon_{\bar{x}}$, $\sigma_{\bar{y}} = E_S\rho_{\bar{y}}\varepsilon_{\bar{y}}$이고, 여기에서 E_s는 철근의 탄성계수이고, $\rho_{\bar{x}}$와 $\rho_{\bar{y}}$는 $\bar{x}$와 $\bar{y}$방향의 철근비이다. 보강철근 재료는 탄성-완전 소성으로 가정 하였다. 강성행렬 계산에 있어서, 수치해석상의 문제를 회피하기 위하여 $\varepsilon_{\bar{x}}$나 $\varepsilon_{\bar{y}}$가 항복변형률을 초과했을 때도 동일한 E_s를 사용하여 계산하였다.

여기에 따른 초과응력은 적당하게 잔여힘을 계산할 때 고려하였다. 6개의 3x3 p-행렬은 Min과 Gupta[20]에 주어 졌으므로 여기에서는 생략한다.

횡방향 전단강성 $[k_{s_{ij}}]$는 아래와 같이 표현할 수 있다.

$$[k_{s_{ij}}]_{6x6} = E_7 h \begin{bmatrix} q_{a_{11}} & \frac{2}{h} q_{a_{12}} \\ \frac{2}{h} q_{a_{21}} & \left(\frac{2}{h}\right)^2 q_{a_{22}} \end{bmatrix} + E_7 h \begin{bmatrix} q_{b_{11}} & \frac{2}{h} q_{b_{12}} \\ \frac{2}{h} q_{b_{21}} & \left(\frac{2}{h}\right)^2 q_{b_{22}} \end{bmatrix}, \quad (3)$$

여기에서 E_7은 횡전단 탄성계수이다. $[q_a]$와 $[q_b]$는 Min과 Gupta[20]에 주어졌다.

원래의 Akbar-Gupta 프로그램에서[3] 균열방향에 평행한 횡전단 탄성계수를 0으로 가정하였다. 이 가정이 수치해석 상에 문제를 야기 시키며[18], 이 문제를 해결하기 위해서는 0이 아닌 어떤 횡전단 탄성계수를 사용하더라도 해결될 수 있을 것이다. 해석결과가 실제로 사용한 값에 예민하지 않을 것으로 사료하여 균열단면에 평행, 그리고 연직한 방향 모두에서 동일한 전단계수로 가정하였다. 식 (3)은 이러한 가정을 함축하고 있다.

강성행렬의 수치적분은 균열 전과 균열 후의 콘크리트요소에 대해서 각각 적당한 선택적인 Gaussian quadrature기법을 사용하였다. 균열이 일어나기 전에는 전단구속을 피하기 위해서 변형률 ε_x와 ε_y에 관계되는 강성행렬은 2x2 quadrature를 사용하였고, 전단변형률 γ_{xy}에 관계되는 항들은 1x1 quadrature를 사용하였다[23]. 균열이 일어난 뒤에는 직접과 전단 응력들과 변형률들이 서로 결부되어서, 앞에서의 선택적 적분기법을 사용할 수 없게 된다. Gupta와 Akbar[11]는 전체 콘크리트 강성행렬을 요소 중심에서 1점으로 적분하는 방법을 제시하였고, 이 때문에 야기되는 singularity는 보강철근 요소를 2x2로 적분함으로써 피하였다.

위에서 다룬 식들은 전체 좌표계를 기준으로 개발되었다. 각각의 노드는 6 자유도, 즉 3 방향의 직선 변위와 3 방향의 회전 변위를 갖고 있다. 그러나, 본 연구에서 다룬 쉘요소는 쉘 좌표계 $\bar{x}$와 $\bar{y}$를 따르는 2 방향의 회전 변위만을 갖고 있다. 따라서 쉘요소는 $\bar{z}$ 좌표를 따라서는 회전 강성이 0이 된다. 만약 어떤 노드에서 곡률이 매우 적거나 0인 경우에 조합된 전체 강성행렬이 singular가 된다. 이 문제를 피하기 위해서 각 노드에서 3 방향의 전체 구조물 좌표계 회전 변위를 쉘 좌표계 $\bar{x}$와 $\bar{y}$를 따르는 2 방향의 회전 변위로 변환하였다. 따라서 변환을 거친 노드는 전체 구조물 좌표계에서 5 자유도가 되는 것이다.

어떤 노드에서 변환행렬은 4개의 작은 행렬로 분류할 수 있다.

$$[T]_{6x5} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} [I]_{3x3} & [0]_{3x2} \\ [0]_{3x3} & [T_{22}]_{3x2} \end{bmatrix}, \quad [T_{22}] = \begin{bmatrix} \bar{x}_{,x} & \bar{y}_{,x} \\ \bar{x}_{,y} & \bar{y}_{,y} \\ \bar{x}_{,z} & \bar{y}_{,z} \end{bmatrix}, \quad (4)$$

여기에서 $[I]$와 $[0]$는 identity와 null 행렬을 각각 표시하며, $\bar{x}_{,x}$는 전체 좌표계 x에 대한 쉘요소 좌표계 $\bar{x}$의 부분 도함수를 표시한다 (즉, $\bar{x}_{,x}$는 $\partial \bar{x} / \partial x$를 의미 한다). 이 변환행렬을 식 (1)의 앞과 뒤에 곱해주면,

$$[\overline{k_{p_{ij}}}]_{5x5} = \begin{bmatrix} h[P] & 0 \\ 0 & [T_{22}]_i^T \frac{h}{3} [P] [T_{22}]_j \end{bmatrix}, \quad (5)$$

이 된다. 식 (3)의 횡방향 전단강성 $[k_{s_{ij}}]$는 또한 아래와 같이 표현할 수 있다.

$$[\overline{k_{s_{ij}}}]_{5x5} = E_7 \begin{bmatrix} h[q_{11}] & 2[q_{12}][T_{22}]_j \\ [T_{22}]_i^T 2[q_{21}] & [T_{22}]_i^T \left(\frac{4}{h}\right) [q_{22}][T_{22}]_j \end{bmatrix}, \quad (6)$$

여기에서

$$\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = \begin{bmatrix} q_{a_{11}}+q_{b_{11}} & q_{a_{12}}+q_{b_{12}} \\ q_{a_{21}}+q_{b_{21}} & q_{a_{22}}+q_{b_{22}} \end{bmatrix}.$$

3. 벡터 알고리즘

벡터화는 벡터연산 장치의 이점을 최대한도로 살리기 위해서 가능한 한 벡터 길이를 구조물의 전체 유한요소 수로 유지하도록 하였다. 매번의 반복계산에서 구성방정식의 값을 제외하고는 불변이므로 계산에 필요한 값들은 한번 계산하여 저장해 놓으면 다시 계산할 필요 없이 반복하여 사용할 수 있다. 앞에서 설명한 바와 같이 콘크리트 요소가 균열 전과 후에 수치적분의 방법이 변할 뿐만 아니라, 균열에 대한 판정이 개개의 요소에 대해서 행해져야 하므로 벡터의 길이를 전체 유한요소 수로 유지하는데는 문제가 있다. 따라서 전체의 강성행렬 계산을 스칼라계산으로 하는 것을 피하기 위해서 매 반복계산에서 먼저 전체 요소에 균열이 발생하지 않았다고 가정하여 백터계산을 하고, 개개의 균열요소에 대해서는 다시 강성행렬 계산을 하여 갱신하였다. 전체구조물에 상대적으로 균열요소의 수가 적을 때는 본 방법이 좋은 결과를 가져올 것이라고 기대할 수 있을 것이다. 요소 강성행렬 계산의 벡터 알고리즘은 Min과 Gupta[20]에서 찾을 수 있다.

4. 알고리즘의 효율평가

벡터 알고리즘의 효율을 평가하기 위해서 2 모델, 쌍곡 '안장' 쉘과 쌍곡 냉각탑을 사용하였다. 쌍곡 '안장' 쉘은 요소의 크기에 변화를 주어 2 모델로 자르고, 쌍곡 냉각탑으로는 하나의 모델을 형성하여 총 요소 수가 벡터 알고리즘의 효율에 미치는 영향을 산정 하였다. 3 모델에 대한 변수는 표 1에 주어 졌다.

표 1: 모델들의 변수

	쌍곡 '안장' 쉘		쌍곡 냉각탑
모델	S16	S32	C24
총 요소 수	89	305	432
총 절점 수	133	389	475

Cray Y-MP를 이용하여 3 모델들에 대한 원래의 Akbar-Gupta 프로그램을 이용한 스칼라 계산과 본 연구의 알고리즘에 의한 벡터 계산에 대한 CPU 시간의 비교가 표 2에 주어 졌다. CPU 시간은 Cray Y-MP의 CPU timer 'SECOND'[5]로 측정하였으며, 컴파일러 지정 시에 스칼라 프로그램은 'novector' 옵션을 지정하였고, 벡터 프로그램은 'full' 옵션으로 지정하였다[4]. Novector-옵션에 비해서 기본설정인 full-옵션은 컴파일러 상에서 최적화하기 때문에 계산 속도의 증가를 가져올 수 있다. Akbar-Gupta 프로그램을 이 옵션으로 계산 했을 때 결과에 오차를 가져와서, novector-옵션으로 계산을 시도하였다. Novector-옵션은 벡터화와 스칼라 최적화를 시도하고, full-옵션은 앞의 옵션에 더하여 완전한 최적화를 시도한다. 실제에 있어서 프로그램을 전혀 벡터화하지 않고 오직 이 novector-옵션을 사용하여 S16모델의 계산을 시도 했을 때 컴파일러 상에서 아무런 시도도 하지 않는 'off' 옵션에 비해서 약 3.8배의 CPU 시간의 속도 증가를 가져 왔다. 본 벡터 알고리즘에 의한

스칼라 계산에서 벡터 계산을 했을 때의 실행속도의 증가는 요소 수가 가장 작은 모델 S16의 1.7배에서 요소 수가 가장 많은 모델 C24에서 7.6배에 도달했다. 실행속도의 증가는 요소 수가 많아 질수록 증가하는 경향이 있으며, 이는 요소 수가 많을수록 벡터의 길이가 길어져서 효율이 증가되는 것으로 설명될 수 있다.

표 2: Cray Y-MP에서의 강성행렬의 스칼라계산과 벡터계산의 비교

모델	반복계산 회수	CPU 시간 (seconds)		속도증가 (스칼라/벡터)
		스칼라계산	벡터계산	
S16	265	4.3	2.5	1.7
S32	493	19.2	6.1	3.1
C24	436	72.9	9.6	7.6

균열 유한요소 수의 전체 유한요소 수에 대한 비율도 효율에 상당한 영향을 미칠 것이라고 짐작할 수 있다. 쌍곡 안장쉘 모델 S32의 요소 수는 305로써 속도 증가는 3.1배이 었으나, 쌍곡냉각탑 *C24*는 *432*의 요소 수로서 속도 증가는 *7.6*배에 도달했다. C24가 S32에 비해서 총 요소 수는 오직 1.4배 증가 했는데, 속도증가 효율은 2.5배 증가하였다. 그 이유로는 쌍곡 안장쉘의 균열요소의 비율이 냉각탑에 비해서 상당히 증가되고 있음을 추론 할 수 있을 것이다[18].

5. 결론

균열이 나타나기 전과 난 뒤의 철근콘크리트 쉘 요소 강성행렬 계산의 벡터 알고리즘을 Cray Y-MP 슈퍼컴퓨터에 적용하여 벡터 알고리즘의 계산속도 증가 효율을 평가하였다. 알고리즘은 전체 요소에 균열이 일어나지 않았다는 가정을 도입하여 벡터의 길이를 유한요소 수로 유지하는 것을 기본으로 하여 개발되었다. 균열 유한요소의 강성행렬 계산은 분리하여 스칼라계산으로 재계산하였다. 3 모델에 의한 비탄성해석을 수행한 결과, 스칼라계산에 대한 벡터계산의 계산속도 증가는 Cray Y-MP 슈퍼컴퓨터에서 1.7배에서 7.6배에 이르렀다. 총 유한요소의 수가 증가할 때, 또는 총 균열요소의 수가 감소할 때 속도증가 효율은 증가하였다.

감사의 글

미국 North Carolina주의 Research Triangle Park시에 있는 North Carolina Supercomputing Center의 Cray Y-MP 슈퍼컴퓨터를 사용한 것에 대해서 깊은 감사의 뜻을 전한다.

참고문헌

[1] ACI 318-89. (1989). Building code requirements for reinforced concrete (*ACI 318-89*), American Concrete Institute, Box 19150, Redford Station Detroit, Michigan 48219.
[2] Ahmad, Sohrabuddin, Irons, Bruce M., and Zienkiewicz, O. C. (1970). "Analysis of thick and thin shell structures by curved finite elements." *Int. J. for Numer. Meth. in Eng.*, 2, 419-451.
[3] Akbar, Habibollah, and Gupta, Ajaya K. (1985). "Membrane reinforcement in concrete

shells: design versus nonlinear behavior." North Carolina State University, Raleigh, North Carolina 27695-7908, January. Reinforced Concrete Shell Research Report.
[4] Cray SR-0018 C, CFT77 reference manual, SR-0018 C, Cray Research, Inc.
[5] Cray SR-2081 6.0, UNICOS math and scientific library reference manual, Cray Research, Inc.
[6] Dongarra, Jack J., and Eisenstat, Stanley C. (1984). "Squeezing the most out of an algorithm in Cray FORTRAN." *ACM Transactions on Mathematical Software*, 10(30), 219-230.
[7] Farhat, C., Sobh, N., and Park, K. C. (1990). "Transient finite element computations on 65536 processors: the connection machine." *Int. J. for Numer. Methods in Eng.*, 30, 27-55.
[8] Foresti, S., Hassanzadeh, S., Murakami, H., and Sonnad, V. (1993) "Parallel rapid operator for iterative finite element solvers on a sheared memory machine," *Parallel Computing*, 19, 1-7.
[9] Golub, Gene and Ortega, James M., (1993). Scientific computing - an introduction with parallel computing. Academic Press, Inc., San Diago,CA 92101-4311
[10] Gupta, A. K. and Mohraz, B., (1973) "A method of computing numerically integrated stiffness matrices." *Int. J. Numer. Methods in Eng.*, 5, 83-89.
[11] Gupta, A. K., and Akbar, H. (1984a). "Cracking in reinforced concrete analysis ." *J. Struct. Engrg.*, ASCE, 110(8), 1735-1746.
[12] Gupta, A. K. (1984b). "Membrane reinforcement in concrete shells:a review." *Nuclear Engineering and Design*, 82, 63-75.
[13] Hand, Frank R., Pecknold, David A., and Schnobrich, William C. (1973). "Nonlinear layered analysis of RC plates and shells." *J. Struct. Div.*, ASCE, 99(7), 1491-1505.
[14] Lambiotte, Jules J. (1975). The Solution of linear systems of equations on a vector computer. Ph.D. thesis, University of Virginia.
[15] Lin, Cheng-Shung, and Scordelis, Alexander C. (1975). "Nonlinear analysis of RC shells of general form." *J. Struct. Div.*, ASCE, 101 (3),523-538.
[16] Milford, R. V., and Schnobrich, W. C. (1984). "Nonlinear behavior of reinforced concrete cooling towers." Technical report, University of Illinois, Urbana-Champaign, Illinois 61801, May. Structural Research Series, No. 514.
[17] Min, Chang Shik, and Gupta, Ajaya K. (1991). "Vector finite-element analysis using IBM 3090-600E VF." *Commun. in Applied Numer. Methods*, Vol. 7, No. 2, 155-164.
[18] Min, Chang Shik, and Gupta, Ajaya K. (1992). "A study of inelastic behavior of reinforced concrete shells using supercomputers." Department of Civil Engrg., North Carolina State Univ., Raleigh, North Carolina 27695-7908, March. Reinforced Concrete Shell Research Report.
[19] Min, Chang Shik and Gupta, Ajaya Kumar. (1993). "Inelastic behavior of hyperbolic cooling tower." *J. of Struct. Engrg.*, ASCE, Vol. 119, No. 7, July, 2235-2255.
[20] Min, Chang Shik and Gupta, Ajaya Kumar. (1994). "Vector algorithm for reinforced concrete shell element stiffness matrix." *Structural Engineering and Mechanics*, Vol. 2, No. 2, 125-139.
[21] Noor, Ahmed K., and Peters, Jeanne M. (1986). "Element stiffness computation on CDC Cyber 205 computer." *Commun. in Applied Numer. Methods*, 2, 317-328.
[22] Ortega, J. M., (1988). Introduction of parallel and vector solutions of linear systems, Plenum Press, New York, N.Y. 10013.
[23] Pawsey, Stuart F., and Clough, Ray W. (1971). "Improved numerical integration of thick shell finite elements." *Int. J. for Numer. Methods in Eng.*, 3, 575-586.
[24] Yagawa, G., Yoshioka, A., Yoshimura, S., and Soneda, N. (1993). "A parallel finite element method with a supercomputer network." *Computers and Structures*, Vol. 47, No. 3, 407-418.