SIAM J. SCI. COMPUT.
Vol. 00, No. 0, pp. 000–000, May 0000

A NEW ADAPTIVE GMRES ALGORITHM FOR ACHIEVING HIGH ACCURACY*

MARIA SOSONKINA†, LAYNE T. WATSON†, AND RAKESH K. KAPANIA‡

Abstract. GMRES(k) is widely used for solving nonsymmetric linear systems. However, it is inadequate either when it converges only for k close to the problem size or when numerical error in the modified Gram-Schmidt process used in the GMRES orthogonalization phase dramatically affects the algorithm performance. An adaptive version of GMRES(k) which tunes the restart value k based on criteria estimating the GMRES convergence rate for the given problem is proposed here. This adaptive GMRES(k) procedure outperforms standard GMRES(k), several other GMRES-like methods, and QMR on actual large scale sparse structural mechanics postbuckling and analog circuit simulation problems. There are some applications, such as homotopy methods for high Reynolds number viscous flows, solid mechanics postbuckling analysis, and analog circuit simulation, where very high accuracy in the linear system solutions is essential. In this context, the modified Gram-Schmidt process in GMRES can fail causing the entire GMRES iteration to fail. It is shown that the adaptive GMRES(k) with the orthogonalization performed by Householder transformations succeeds whenever GMRES(k) with the orthogonalization performed by the modified Gram-Schmidt process fails, and the extra cost of computing Householder transformations is justified for these applications.

Key words. GMRES method, Krylov subspace methods, nonsymmetric linear systems, homotopy curve tracking, sparse matrix

AMS(MOS) subject classifications. 65F10, 65F50, 65H10

1. Introduction.

Computational challenges of the contemporary world make iterative methods significant "team players" in a sophisticated problem solving process, the overall success of which largely depends on linear system solutions. In the context of homotopy zero curve tracking algorithms [21], for example, to locate the difficult parts of the curve (sharp turning points) a linear system solution in the correction phase of the stepping algorithm along the curve has to be very accurate. In this paper, the ability of the generalized minimal residual algorithm (GMRES) [16] to deal successfully with matrices of varying difficulty and to give high accuracy solutions is investigated.

High accuracy computation is known to be especially important for some circuit design and simulation problems, which involve nonsymmetric unstructured matrices [14]. These problems have proven to be very difficult for any iterative method, and presently, only specially tailored direct methods are used to solve them in a production environment. The impact of numerical error is shown here to be significant in the postbuckling stability analysis of structures which exhibit snap-back and snap-through phenomena. A postbuckling analysis amounts to tracking what is called an equilibrium curve, involving the solution of linear systems with (tangent stiffness) matrices that vary along the equilibrium curve. Mathematically, the snap-back and snap-through behavior of structures corresponds to symmetric, structured, and strongly indefinite

* Received by the editors ; accepted for publication (in revised form) .

† Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061-0106. The work of these authors was supported in part by Department of Energy Grant DE-FG05-88ER25068 and Air Force Office of Scientific Research Grant F49620-92-J-0236.

‡ Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061-0203.

tangent stiffness matrices, which can be difficult for Krylov subspace iterative methods. Along some portions of the curve the tangent stiffness matrices may be well conditioned and positive definite—easy for iterative methods. This wide variation in linear system difficulty clearly suggests an adaptive strategy. With good preconditioning, the details of an iterative solver become less important, and for many real problems good preconditioning strategies (e.g., multigrid) are available. For some classes of problems, like analog DC circuit design and structural postbuckling stability analysis, good preconditioning strategies are not known, and subtle implementation details of iterative algorithms become paramount.

Section 2 develops the proposed subspace enlarging strategy for GMRES(k) and addresses some technical issues of the GMRES implementation. Section 3 describes algorithms and realistic test problems for numerical experiments. The numerical results are presented and discussed in §4, and §5 concludes.

2. An implementation of the adaptive GMRES(k) algorithm. The GMRES algorithm [16] is used for solving a linear system $Ax = b$ with an $n \times n$ nonsymmetric invertible coefficient matrix A. Similar to the classical conjugate gradient method, GMRES produces approximate solutions x_j which are characterized by a minimization property over the Krylov subspaces $K(j, A, r_0) \equiv \operatorname{span}\{r_0,\ Ar_0,\ A^2 r_0,\ \cdots,\ A^{(j-1)} r_0\}$, where $r_0 = \|b - Ax_0\|$ and j is the iteration number. However, unlike the conjugate gradient algorithm, the work and memory required by GMRES grow proportionately to the iteration number since GMRES needs all j vectors to construct an orthonormal basis of $K(j, A, r_0)$. In practice, the restarted version GMRES(k) is used, where the algorithm is restarted every k iterations, taking x_k as the initial guess for the next cycle of k iterations, until the residual norm is small enough. Pseudocode for GMRES(k) is:

choose x, tol
$r := b - Ax;$
while $\|r\| > tol$ **do**
begin
  $v_1 := r/\|r\|;$
  for $j = 1$ **step** 1 **until** k **do**
  begin
    for $i = 1$ **step** 1 **until** j **do** $h_{i,j} := (Av_j)^T v_i;$

$$\tilde{v}_{j+1} := Av_j - \sum_{i=1}^{j} h_{i,j} v_i;$$

    $h_{j+1,j} := \|\tilde{v}_{j+1}\|;$
    $v_{j+1} := \tilde{v}_{j+1} \;/\; h_{j+1,j};$
    update $\|r\|$ as described in [16];
    if $\|r\| \leqq tol$ **then goto** L1
  end
L1: $e_1 := (1, 0, \cdots, 0)^T;$
  solve $\min_y \big\| \|r\| e_1 - \bar{H}_j y \big\|$ for y_j where $\bar{H}_j$ is described in [16];
  $V_j := [v_1, \cdots, v_j];$ $x := x + V_j y_j;$ $r := b - Ax$
end

In practice, the modified version of the Gram-Schmidt process is used in the construction of an orthonormal basis for the Krylov subspace. The disadvantage of the restarted version is that it may stagnate and never reach the solution. The essence of the adaptive GMRES strategy proposed here is to adapt the parameter k to the problem, similar in spirit to how a variable order ODE algorithm tunes the order k. With FORTRAN 90, which provides pointers and dynamic memory management, dealing with the variable storage requirements implied by varying k is not too difficult. The parameter k can be both increased and decreased—an increase-only strategy is described next followed by pseudocode.

If k in GMRES(k) is not sufficiently large, GMRES(k) can stagnate. A test of stagnation developed in [3] detects an insufficient residual norm reduction in the restart number k of steps by estimating the GMRES behavior on a particular linear system. Precisely, GMRES(k) is declared to have stagnated and the iteration is aborted if at the rate of progress over the last restart cycle of steps, the residual norm tolerance cannot be met in some large multiple (bgv) of the remaining number of steps allowed ($itmax$ is a bound on the number of steps permitted). Slow progress of GMRES(k) which indicates that an increase in the restart value k may be beneficial [18] can be detected with a similar test. The near-stagnation test uses a different, smaller multiple (smv) of the remaining allowed number of steps. If near-stagnation occurs, the restart value k is incremented by some value m and the *same* restart cycle continues. Restarting would mean repeating the nonproductive iterations that previously resulted in stagnation, at least in the case of complete stagnation (no residual reduction at all). Such incrementing is used whenever needed if the restart value k is less than some maximum value $kmax$. When the maximum value for k is reached, adaptive GMRES(k) proceeds as GMRES($kmax$). The values of the parameters smv, bgv, and m are established experimentally and can remain unchanged for most problems.

The convergence of GMRES may also be seriously affected by roundoff error, which is especially noticeable when a high accuracy solution is required. The orthogonalization phase of GMRES is susceptible to numerical instability. Let Q be a matrix whose columns are obtained by orthogonalizing the columns of a matrix M, and define the error matrix $E = Q^TQ - I$. The error matrices E_{MGS}, E_{HR} using the modified Gram-Schmidt and Householder reflection methods, respectively, to construct Q from M satisfy [2]

$$\|E_{MGS}\|_2 \sim \mathbf{u}\,\mathrm{cond}(M), \qquad \|E_{HR}\|_2 \sim \mathbf{u},$$

where $\mathbf{u}$ is the machine unit roundoff. Clearly the orthogonalization with Householder reflections is more robust. An implementation of GMRES(k) using Householder reflections is given in [19]. In theory, the implementation of GMRES using Householder reflections is about twice as expensive as when modified Gram-Schmidt is used [20]. However, the Householder reflection method produces a more accurate orthogonalization of the Krylov subspace basis when the basis vectors are nearly linearly dependent and the modified Gram-Schmidt method fails to orthogonalize the basis vectors; this can result in fewer GMRES iterations compensating for the higher cost per iteration using Householder reflections. Let e_j be the jth standard basis vector and all norms the 2-norm. Pseudocode for an adaptive version of GMRES(k) with orthogonalization via Householder reflections implemented as in [19] and [20] follows. Call this algorithm AGMRES(k).

choose $x, itmax, kmax, m$;
$r := b - Ax$; $itno := 0$; $cnmax := 1/(50\mathbf{u})$;
$xtol := \max\{100.0, 1.01avnz\}\mathbf{u}$; $tol := \max\{\|r\|, \|b\|\}xtol$;
while $\|r\| > tol$ **do**
begin
  $r^{old} := r$;
  determine $P_1 r = \pm\|r\|e_1$
    where the Householder transformation matrix P_1 is defined in [8];
  for $j := 1$ **step** 1 **until** k **do**
  begin
L1: $itno := itno + 1$;
    $v := P_j \cdots P_1 A P_1 \cdots P_j e_j$;
    determine P_{j+1} such that $P_{j+1}v$ has zero components
      after the $(j+1)$st;
    update $\|r\|$ as described in [16];
    compute incremental condition number ICN as in [1];
    if $ICN > cnmax$ **then** abort;
    if $\|r\| \leqq tol$ **then goto** L2
  end
  $test := k \times \log\left[tol/\|r\|\right] / \log\left[\|r\|/\left((1.0 + 10\mathbf{u})\,\|r^{old}\|\right)\right]$;
  if $k \leqq kmax - m$ **and** $test \geqq smv \times (itmax - itno)$ **then**
    $k := k + m$;
    goto L1
  end if
L2: $e_1 := (1, 0, \cdots, 0)^T$;
  solve $\min\limits_y \left\| \|r\|e_1 - \bar{H}_j y \right\|$ for y_j where $\bar{H}_j$ is described in [16];
  $q := \begin{pmatrix} y_j \\ 0 \end{pmatrix}$;
  $x := x + P_1 \cdots P_j q$; $r := b - Ax$;
  if $\|r\| \leqq tol$ **then** exit;
  if $\|r^{old}\| < \|r\|$ **then**
    if $\|r\| < tol^{2/3}$ **then**
      exit
    else
      abort
    end if
  end if
  $test := k \times \log\left[tol/\|r\|\right] / \log\left[\|r\|/\left((1.0 + 10\mathbf{u})\,\|r^{old}\|\right)\right]$;
  if $test \geqq bgv \times (itmax - itno)$ **then**

abort

end if

end

The rounding error of a sparse matrix-vector multiplication depends on only the nonzero entries in each row of the sparse matrix, so the error tolerance $xtol$ is proportional to the average number of nonzeros per row $avnz =$ (number of nonzeros in $A)/n$. The code from [20] assumes that the initial estimate $x_0 = 0$, which is certainly not the case for homotopy zero curve tracking, where the previous tangent vector provides a good estimate x_0. Since GMRES convergence is normally measured by reduction in the initial residual norm, the convergence tolerance is $tol = \max\{\|r_0\|, \|b\|\}xtol$.

A possible symptom of AGMRES(k) going astray is an increase in the residual norm between restarts (the residual norm is computed by direct evaluation at each restart). If the residual norm on the previous restart is actually smaller than the current residual norm, then AGMRES(k) terminates. The solution is considered acceptable if $\|r\| < tol^{2/3}$, although this loss of accuracy may cause the client algorithm (the "outer" algorithm requiring solutions to linear systems) to work harder or fail. A robust client algorithm can deal gracefully with a loss of accuracy in the linear system solutions. If $\|r\| \geqq tol^{2/3}$, AGMRES(k) is deemed to have failed. In this latter case, the continuation of GMRES(k) would typically result in reaching a limit on the number of iterations allowed and a possible repetition of $\|r^{old}\| < \|r\|$ in later restarts. AGMRES(k) may exceed an iteration limit when it is affected by roundoff errors in the case of a (nearly) singular GMRES least-squares problem. The condition number of the GMRES least-squares problem is monitored by the incremental condition estimate [1] as in [3]. AGMRES(k) terminates when the estimated condition number is greater than $1/(50\mathbf{u})$.

3. Numerical experiments. A comparative study of the proposed adaptive version of GMRES(k) considers some other versions of GMRES(k): the standard implementation of GMRES(k) [16], the adaptive GMRES(k) with modified Gram-Schmidt orthogonalization, GMRES(k) with GMRES-ILU preconditioning [15], and the popular iterative method QMR [6]. A brief description of the GMRES(k) with GMRES-ILU preconditioning and QMR algorithms follows.

The *GMRES(k) with GMRES-ILU preconditioning* method is a variation of flexible GMRES developed in [15]. Flexible GMRES(k) differs from the standard preconditioned GMRES(k) implementation by allowing variations in preconditioning at each iteration. Let M_i be the computed preconditioner at the ith iteration. Then an iterate x_k is given by

$$x_k = x_0 + [M_1 v_1, \cdots, M_k v_k]\, y_k,$$

where y_k is the solution to the GMRES least-squares problem and the v_i are the same as in the standard GMRES(k). Here the application of M_i corresponds to an "inner" iteration consisting of $\tilde{k} \ll k$ steps of an ILU preconditioned GMRES and with the iteration limit $\widetilde{itmax} \ll itmax$.

The *quasi-minimal residual* method QMR [6] is based on a modification of the classical nonsymmetric Lanczos algorithm. Given two starting vectors v_1 and w_1, the Lanczos algorithm uses three-term recurrences to generate sequences $\{v_i\}_{i=1}^L$ and $\{w_i\}_{i=1}^L$ of vectors satisfying

$$\begin{aligned}\operatorname{span}\{v_1,\cdots,v_m\} &= \operatorname{span}\{v_1, Av_1,\cdots,A^{m-1}v_1\},\\ \operatorname{span}\{w_1,\cdots,w_m\} &= \operatorname{span}\{w_1, A^Tw_1,\cdots.(A^T)^{m-1}w_1\},\end{aligned}$$

for $m = 1,\cdots,L$, and $w_i^Tv_j = d_i\delta_{ij}$, with $d_i \neq 0$, for all $i,j = 1,\cdots,L$, where δ_{ij} is the Kronecker delta.

An implementation of QMR based on the look-ahead Lanczos process avoids most breakdowns associated with Lanczos-type algorithms [5]. At the kth iteration, QMR computes an iterate x_k as

$$x_k = x_0 + V^{(k)}z,$$

where the columns of the matrix $V^{(k)}$ are the right look-ahead Lanczos vectors $v_1,\cdots,v_k$, and the vector z is the unique minimizer of

$$\left|\left| ||r_0||e_1 - H_e^{(k)}z \right|\right|,$$

where $H_e^{(k)}$ is a $k\times k$ block tridiagonal matrix augmented with a row of the form ρe_k^T.

Based on the conclusions of [4], [10], and [12] *incomplete LU factorization* with zero fill in (ILU(0)) and *Gill-Murray* preconditioning are considered for nonsymmetric unstructured and symmetric skyline structured matrices, respectively. Let $Z \subset \{(i,j) \mid |1 \leqq i,j \leqq n,\ i\neq j\}$ be a subset of the set of indices where the matrix A has zeros. Then the ILU factorization of A is given by $Q = LU$, where L and U are lower triangular and unit upper triangular matrices, respectively, satisfying

$$\begin{aligned}&L_{ij} = U_{ij} = 0, && \text{for } (i,j)\in Z;\\ &Q_{ij} = A_{ij}, && \text{for } (i,j)\notin Z, i\neq j;\\ &Q_{ii} = A_{ii}, && \text{whenever possible.}\end{aligned}$$

The factors of the preconditioning matrix Q preserve the sparsity pattern of A. The Gill-Murray preconditioner [7] for a matrix A with a skyline sparsity pattern is a positive definite approximation $Q = GG^T$ to A. The lower triangular matrix G has the same skyline structure as A, and is chosen such that G is always well conditioned.

For future reference the considered algorithms will be called:

AGH — adaptive restarted GMRES with Householder reflections in the orthogonalization phase;

AGS — adaptive restarted GMRES with modified Gram-Schmidt in the orthogonalization phase;

FGS — flexible restarted GMRES with $\tilde{k} = k/3$, $\widetilde{itmax} = itmax/20$;

GS — standard implementation of restarted GMRES;

Q — three-term recursion QMR.

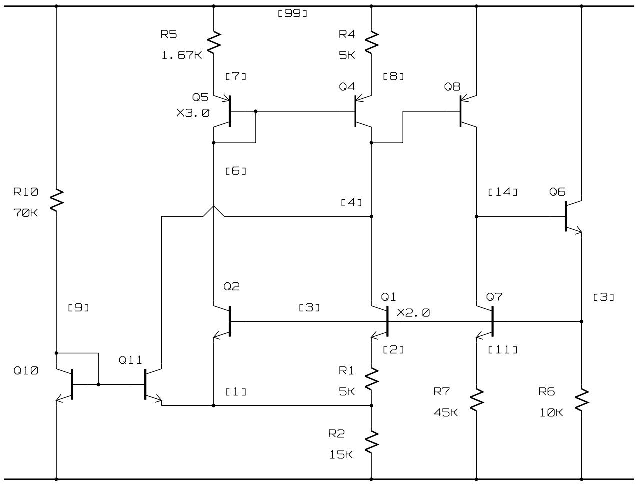

FIG. 1. *Circuit diagram for the circuit "vref".*

3.1. Circuit design and simulation. The circuit simulation test problems considered here are from [12] and [14], and are representative of full-scale commercial design at AT&T Bell Laboratories. The DC operating point problem is typically formulated as a system of n equations in n unknowns, i.e., $F(x) = 0$, where $F : E^n \to E^n$, and E^n is n-dimensional real Euclidean space. The unknowns are voltages and branch currents and the equations represent the application of Kirchoff's current and voltage laws at various points in the circuit. Different circuit components (resistor, transistor, diode, etc.) impose linear and nonlinear relations among the unknowns. In present fabrication technologies, circuit components are arranged in the form of a graph that is almost planar, which limits the density of interconnections between circuit elements, leading to a sparse Jacobian matrix. Figure 1 shows one of the circuits (Brokaw voltage reference [14], "vref") from which the test problems were derived.

Except for rather special cases, circuit equations lead to large nonsymmetric unstructured matrix problems. In some cases, a circuit element like a transistor is replicated as a small "subcircuit", which is installed in place of every transistor in the network. This policy results in a replication of structurally identical submatrices throughout the overall system Jacobian matrix. The sparsity pattern for the Jacobian matrix corresponding to one of the test circuits (bgatt) with the diagonal blocks corresponding to the transistor subcircuits is shown in Fig. 2.

For each test problem, the various iterative methods were applied to a sequence of Jacobian matrices obtained along a homotopy zero curve for that problem. For the rest of the test problems described below, the iterative methods were actually implemented as subroutines within the homotopy curve tracking software HOMPACK [23]. Specifically a FORTRAN 90 version of the normal flow subroutine FIXPNS was used, resulting in the $(n + 1) \times (n + 1)$ linear systems described in [10]—note that these are slightly different from the linear systems for the sparse normal flow algorithm given in [23].

3.2. Space truss stability analysis. The stability characteristics of a space truss under multiple independent loads are described in [9]. A brief derivation of the nonlinear equilibrium equations, whose solution by a homotopy method generates the linear systems to be solved, is given here.

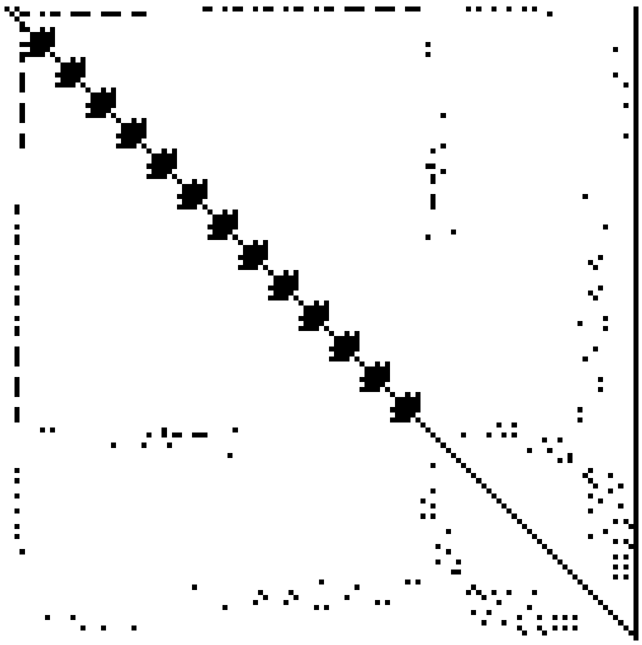

FIG. 2. *Jacobian matrix sparsity pattern corresponding to the circuit "bgatt".*

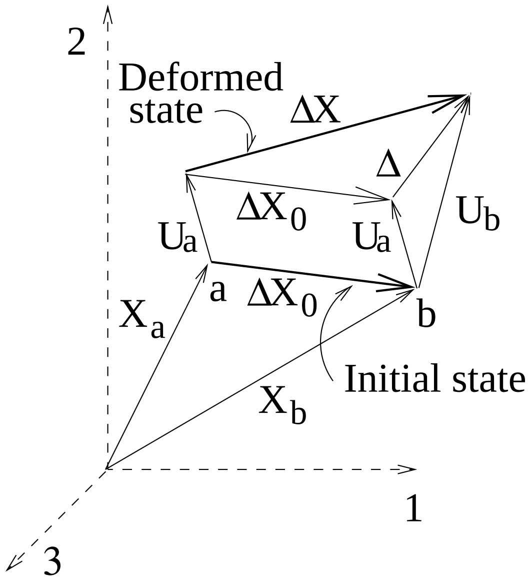

FIG. 3. *Space truss element.*

Finite element model. The axial deformation of the truss element in Fig. 3 is $e = L - L_0$, where L_0 and L are the element lengths in the initial and deformed states, respectively. It follows from Fig. 3 that

$$L = \|\mathbf{\Delta}X\| = \left(\sum_{j=1}^{3} L_j^2\right)^{1/2}, \qquad L_0 = \|\mathbf{\Delta}X_0\| = \left(\sum_{j=1}^{3} L_{0j}^2\right)^{1/2},$$

$$L_j = L_{0j} + \Delta_j, \quad \mathbf{\Delta}X_0 = X_b - X_a, \quad \Delta = U_b - U_a, \quad \mathbf{\Delta}X = \mathbf{\Delta}X_0 + \Delta,$$

where X_a, X_b are the initial coordinate vectors at the a, b-end of the element; U_a, U_b are the global displacement vectors at the a, b-end of the element; and the vectors

$\mathbf{\Delta} X_0$, $\mathbf{\Delta} X$ coincide with the initial, deformed state of the element. The strain energy of the element is

$$\pi = \frac{1}{2}\gamma_0 e^2,$$

where $\gamma_0 = EA/L_0$ is the extensional stiffness of the element, E is Young's modulus of elasticity, and A is the cross sectional area of the truss element. The principle of virtual work yields the equation of equilibrium

$$F(q, \lambda, \bar{Q}) = -\lambda\bar{Q} + \sum_{i=1}^{m} R^i = 0,$$

where q is the generalized displacement vector, λ is the scalar load parameter, $\bar{Q}$ is the load distribution vector, the vector function F is the force imbalance, and

$$R_k^i = \frac{\partial \pi^i}{\partial q_k}, \quad \text{for } k = 1, 2, \cdots, n.$$

The superscript i identifies the m elements of the assemblage. The $n \times (n+1)$ Jacobian matrix becomes

$$DF = \left[K,\ -\bar{Q}\right], \quad \text{where } K = \sum_{i=1}^{m} K^i, \quad K^i = \left[\frac{\partial^2 \pi^i}{\partial q_k \partial q_l}\right].$$

K^i is the tangent stiffness matrix of the element i expressed relative to the generalized displacements of the assemblage. It is computed by the code number method [22] from the global stiffness matrix of element i.

Structure 1. Figure 4 shows a 21-degree-of-freedom lamella dome. The joints lie on the surface of a spherical cap with a radius of 157.25 inches. The support ring of the cap has a radius of 50 inches. All joints are located so that the structure is symmetric about the global x_1-axis which extends down vertically through the apex. The six support joints are fixed in all three directions. Each member of the structure has a cross-sectional area of 0.18 square inches and an elastic modulus of 2900 ksi. The structure is subjected to a vertical load at the apex.

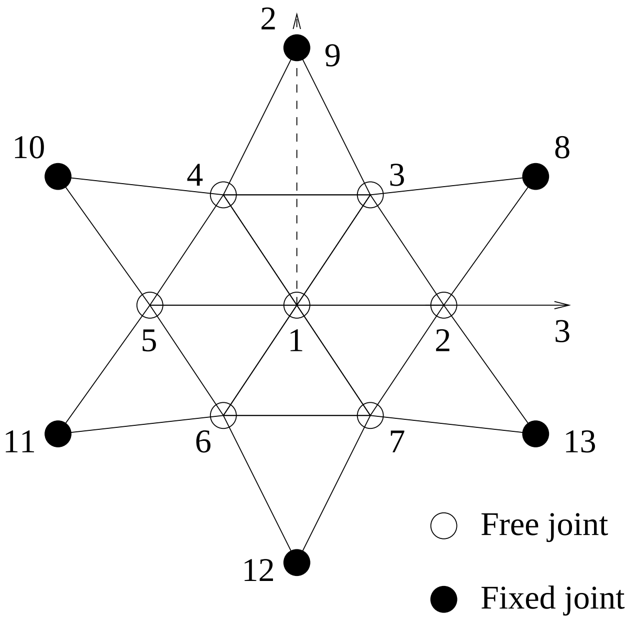

Fig. 4. *Lamella dome* (21*-degree-of-freedom*).

Structure 2. This is another lamella dome with a tension ring at the base having 69 degrees of freedom, similar to Structure 1 but with a different geometrical arrangement of the free joints, different joint constraints, and an unsymmetric loading pattern [9]. The effect of modeling a space truss with truss elements in concentric rings is that changing the number of truss elements changes the model and its behavior. Thus the dome problems with different degrees of freedom considered here are qualitatively different, with different buckling loads.

3.3. Shallow shell stability analysis. Another test problem is the finite element analysis of a shallow cylindrical shell (Fig. 5) under a concentrated load [24]. The shell shown in Fig. 5 is hinged at the longitudinal edges and free along the curved boundaries.

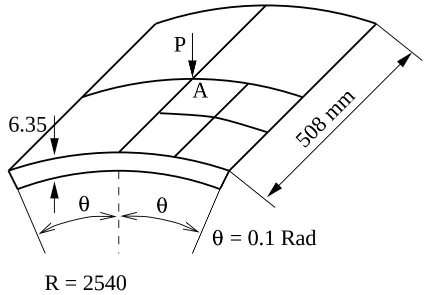

FIG. 5. *Cylindrical shell geometry; load applied at A.*

The structure exhibits snap-through as well as snap-back phenomena, with horizontal and vertical tangents. A finite element model based on a curved quadrilateral 48-degree-of-freedom thin-shell element (Fig. 6) is used to form the equations of equilibrium [11]. The element has four corner nodes and each node has 12 degrees of freedom. Figure 6 shows the undeformed middle surface of the shell element embedded in a fixed Cartesian coordinate system x^i $(i = 1, 2, 3)$. The Cartesian coordinates x^i $(i = 1, 2, 3)$ of the middle surface are modeled by polynomial functions of the curvilinear coordinates ξ and η with a total of N terms:

$$x^i(\xi, \eta) = \sum_{j=1}^{N} C^i_j \xi^{m_j} \eta^{n_j},$$

where the constants m_j and n_j define the powers of ξ and η, respectively, for the jth term. The constants C^i_j are solved for based on the coordinates x_j at N selected points. Precisely, bicubic Hermite polynomials in the curvilinear coordinates ξ and η are used as basis functions for the interpolating polynomial.

The Cartesian coordinates of a given point are described by a system of parametric equations

$$x^i = f^i(\theta^1, \theta^2),$$

where the parameters θ^α $(\alpha = 1, 2)$ serve as coordinates on the surface and can be regarded as an arbitrary selected system.

The two base vectors a_1 and a_2 are obtained as derivatives of f with respect to each component θ^α:

$$a_\alpha = \frac{\partial f^i}{\partial \theta^\alpha} e_i = f^i_\alpha e_i,$$

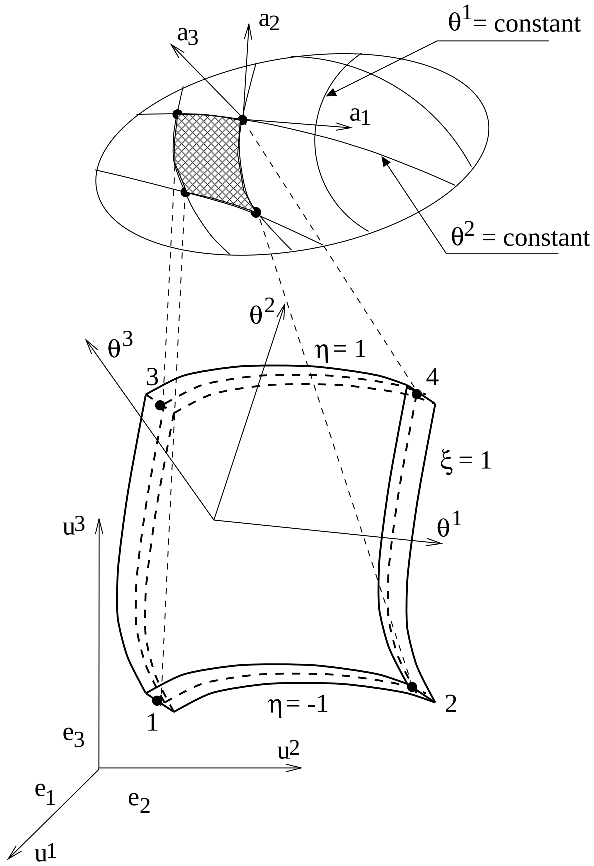

FIG. 6. *Thin-shell element coordinate system.*

where the e_i are the standard basis vectors, and tensor notation is being employed. The unit normal a_3 to the surface is given as

$$a_3 = \frac{a_1 \times a_2}{|a_1 \times a_2|} = n^i e_i,$$

where $n^i = ce_{ijk} f_1^j f_2^k$, e_{ijk} is the alternating symbol, $c = 1/|a_1 \times a_2| = 1/\sqrt{a}$,

$$a = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \qquad a_{\alpha\beta} = a_\alpha \cdot a_\beta = f_\alpha^i f_\beta^i.$$

The metric tensor $a_{\alpha\beta}$ yields the first fundamental form of a surface $ds^2 = a_{\alpha\beta} d\theta^\alpha d\theta^\beta$, where ds is the distance between two neighboring points at θ^α and $\theta^\alpha + d\theta^\alpha$ ($\alpha = 1$, 2). The curvature tensor $b_{\alpha\beta}$ is given as

$$b_{\alpha\beta} = n^i \frac{\partial^2 f^i}{\partial \theta^\alpha \partial \theta^\beta} = n^i f_{\alpha\beta}^i.$$

After deformation, the same point has Cartesian coordinates $\bar{f}^i = f^i + u^i$, where u^i ($i = 1$, 2, 3) are the Cartesian components of the displacement vector. The 12 degrees of freedom for each node are u^i, u_1^i, u_2^i, u_{12}^i ($i = 1$, 2, 3). In the deformed state, the metric and curvature tensors take the form

$$\bar{a}_{\alpha\beta} = (f^i + u^i)_\alpha (f^i + u^i)_\beta = \bar{f}_\alpha^i \bar{f}_\beta^i, \qquad \bar{b}_{\alpha\beta} = \bar{n}^i \bar{f}_{\alpha\beta}^i,$$

where

$$\bar{n}^i = (\bar{a})^{-1/2} e_{ijk} \bar{f}_1^j \bar{f}_2^k, \qquad (\bar{a})^{-1/2} \approx (a)^{-1/2}(1 - A/a),$$
$$A = (\varepsilon_{11} a_{22} + \varepsilon_{22} a_{11} - 2\varepsilon_{12} a_{12}).$$

The tangential strain measure $\varepsilon_{\alpha\beta}$ is given as

$$\varepsilon_{\alpha\beta} = \frac{1}{2}(\bar{a}_{\alpha\beta} - a_{\alpha\beta}) = \frac{1}{2}\left(f_\alpha^i u_\beta^i + f_\beta^i u_\alpha^i + u_\alpha^i u_\beta^i\right), \qquad i = 1, 2, 3.$$

The curvature strain measure $\kappa_{\alpha\beta}$ is given as

$$\kappa_{\alpha\beta} = -\left(\bar{b}_{\alpha\beta} - b_{\alpha\beta}\right) + \frac{1}{2}\left(b_\alpha^\delta \varepsilon_{\beta\delta} + b_\beta^\delta \varepsilon_{\alpha\delta}\right),$$

where b_α^δ is defined in terms of the contravariant tensor $a^{\beta\delta}$ by $b_\alpha^\delta = b_{\alpha\beta} a^{\beta\delta}$.

The total potential energy of the model can be written as

$$\Pi_p(u) = \int\!\!\int_A \left[W(\varepsilon_{\alpha\beta}(u), \kappa_{\alpha\beta}(u)) - p^i u^i\right] \sqrt{a}\, d\theta^1 d\theta^2,$$

where W is the strain energy density per undeformed middle surface unit of the shell, p^i ($i = 1, 2, 3$) are the Cartesian components of the externally applied load, and u^i ($i = 1, 2, 3$) are the Cartesian components of the displacement vector. The strain energy W is given by

$$W = \frac{1}{2} H^{\alpha\beta\lambda\mu} \left(\varepsilon_{\alpha\beta}\varepsilon_{\lambda\mu} + \frac{h^2}{12}\kappa_{\alpha\beta}\kappa_{\lambda\mu}\right),$$

where h is the thickness of the shell. The tensor of elastic moduli $H^{\alpha\beta\lambda\mu}$ is defined as

$$H^{\alpha\beta\lambda\mu} = \frac{Eh}{2(1+\nu)}\left[a^{\alpha\lambda}a^{\beta\mu} + a^{\alpha\mu}a^{\beta\lambda} + \frac{2\nu}{1-\nu}a^{\alpha\beta}a^{\lambda\mu}\right],$$

where E is Young's modulus, ν is Poisson's ratio, $a^{\alpha\beta} a_{\lambda\beta} = \delta_{\alpha\lambda}$ defines the contravariant tensor $a^{\alpha\beta}$, and $\delta_{\alpha\lambda}$ is the Kronecker symbol.

The equilibrium equations of the model are obtained by setting the variation $\delta\Pi_p(u)$ to zero, or equivalently

$$\nabla\, \Pi = 0.$$

The Jacobian matrix of the equilibrium equations is obtained by finite difference approximations.

By symmetry, only a quarter of the shell is modeled using a 2×2 mesh. The material properties are $\nu = 0.3$ and $E = 3.101$ kN/mm^2. An initial external concentrated load of 0.2 kN is applied. The equilibrium curve (load factor λ vs. displacement at the point A) of the cylindrical shell is shown in Fig. 7, where the actual load is 0.2λ kN.

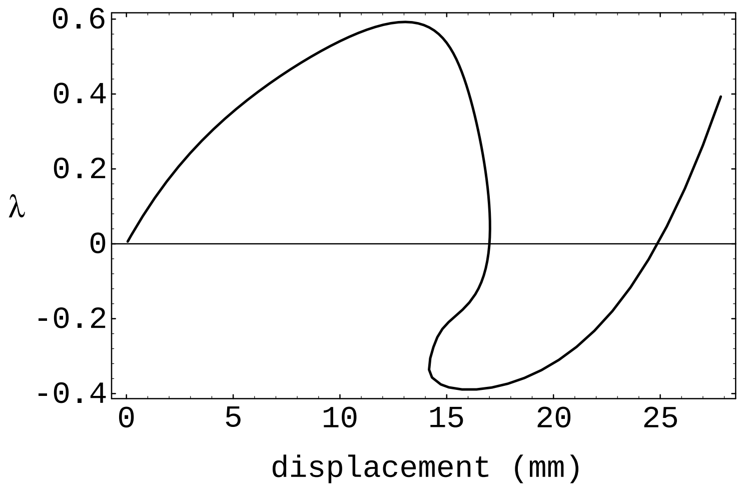

FIG. 7. *Load factor λ vs. displacement u^3 at the point A for the shell in Fig. 5.*

4. Results. Tables 1–6 show the iteration counts and timing results for adaptive GMRES(k) with modified Gram-Schmidt and Householder reflection orthogonalization procedures (AGS, AGH respectively) and different subspace increment values m, standard GMRES(k) (GS), flexible GMRES(k) (FGS), and QMR (Q). The notation AGH:2, for example, means AGH with $m = 2$. Each of the iterative methods was applied to all of the test problems. An iterative solver is deemed to have converged when the relative residual norm reduction is less than $\max\{100, avnz\}\mathbf{u}$. The limit on the number of iterations is $30n$. This number is quite reasonable for realistic problems to be solved on modern computer hardware. This iteration limit requires explanation. First, ideally convergence should occur in far fewer iterations than n, and for well preconditioned PDE problems this usually occurs. However, for other classes of problems and for mediocre preconditioners, more than n iterations may be required, and permitting $30n$ iterations is preferable to permitting very large subspace dimensions k or permitting failure of the linear system solution server to return an accurate solution to the client homotopy algorithm. Second, low accuracy might be acceptable if only Newton corrections were being computed (although inaccurate Newton steps will wreck havoc on homotopy path step size control algorithms), but inaccurate tangents may result in losing the path or in very small, inefficient steps near extremely sharp turns of the homotopy path. Clearly high accuracy is not required *everywhere* along the homotopy path, and effectively adjusting the linear system solution accuracy along the homotopy path, while preserving reliable path step size control, is an open research question.

All reported CPU times are for 64-bit IEEE arithmetic on a DEC Alpha 3000/600 workstation. The average, maximum, and minimum number of iterations per linear system solution along the homotopy zero curve are shown in Tables 1–3. Note that in the FGS column the numbers for the x_k *outer* iterations of flexible GMRES(k) are given. Tables 5, 6 report the total CPU time in seconds required for an iterative method to obtain all the linear system solutions needed for HOMPACK to track a homotopy zero curve for a certain arc length specified for each problem. For the circuit problems (Table 4), the reported CPU time is the time necessary to solve a certain number of linear systems arising along a portion of the homotopy zero curve.

An asterisk denotes convergence failure, meaning any of the following occurred: (1) desired error tolerance not met after $30n$ iterations, (2) dangerous near singularity detected for the GMRES least squares problem, (3) residual norm increased between

Table 1

Average, maximum, and minimum number of iterative solver iterations per linear system along homotopy zero curve for circuit design problems (ILU preconditioner).

n	AGH:2	AGH:4	AGH:6	AGS:2	AGS:4	AGS:6
31	23,112,1 $(2 \rightarrow 6)$	19,108,1 $(2 \rightarrow 6)$	19,108,1 $(2 \rightarrow 6)$	24,112,1 $(2 \rightarrow 6)$	19,108,1 $(2 \rightarrow 6)$	19,108,1 $(2 \rightarrow 6)$
59	52,231,14 $(4 \rightarrow 6)$	48,231,14 $(4 \rightarrow 8)$	49,231,14 $(4 \rightarrow 10)$	53,231,14 $(4 \rightarrow 6)$	53,231,14 $(4 \rightarrow 8)$	*
67	587,1150,344 $(5 \rightarrow 15)$	557,1050,339 $(5 \rightarrow 15)$	471,734,286 $(5 \rightarrow 15)$	576,1578,291 $(5 \rightarrow 15)$	613,1520,330 $(5 \rightarrow 15)$	531,1295,312 $(5 \rightarrow 15)$
125	143,193,84 $(4 \rightarrow 6)$	121,169,84 $(4 \rightarrow 8)$	102,131,82 $(4 \rightarrow 10)$	146,230,84 $(4 \rightarrow 6)$	122,156,84 $(4 \rightarrow 8)$	110,140,73 $(4 \rightarrow 10)$
468	623,1842,252 $(15 \rightarrow 15)$	623,1842,252 $(15 \rightarrow 15)$	623,1842,252 $(15 \rightarrow 15)$	603,1740,252 $(15 \rightarrow 15)$	603,1740,252 $(15 \rightarrow 15)$	603,1740,252 $(15 \rightarrow 15)$
1854	2719,4340,1173 $(35 \rightarrow 47)$	2768,4234,956 $(35 \rightarrow 47)$	2419,3953,874 $(35 \rightarrow 47)$	2710,4165,1347 $(35 \rightarrow 47)$	2725,3990,1096 $(35 \rightarrow 47)$	2523,3947,1060 $(35 \rightarrow 47)$

n	GS	FGS	Q
31	(2) * (6)12,48,1	(2) * (3)8,196,1	*
59	(4) * (10)9,10,7	*	*
67	(5) * (15)288,359,217	(3) * (8)20,107,9	*
125	(4) * (10)83,110,49	(4) * (6)62,142,9	*
468	(15)603,1740,252	*	*
1854	(35) * (47)611,838,539	*	*

Table 2

Average, maximum, and minimum number of iterative solver iterations per linear system along homotopy curve for thin shell problem (Gill-Murray preconditioner).

n	AGH:2	AGH:4	AGH:6	AGS:2	AGS:4	AGS:6	GS	FGS	Q
55	80,1239,1 $(8 \rightarrow 20)$	57,888,1 $(8 \rightarrow 20)$	44,748,1 $(8 \rightarrow 24)$	*	*	*	*	*	*
119	6,134,1 $(5 \rightarrow 5)$	6,134,1 $(5 \rightarrow 5)$	6,134,1 $(5 \rightarrow 5)$	6,140,1 $(5 \rightarrow 5)$	6,140,1 $(5 \rightarrow 5)$	6,140,1 $(5 \rightarrow 5)$	6,140,1 (5)	*	*
1239	3,14,1 $(12 \rightarrow 20)$	3,14,1 $(12 \rightarrow 20)$	3,14,1 $(12 \rightarrow 20)$	*	*	*	*	*	*

Table 3

Average, maximum, and minimum number of iterative solver iterations per linear system along homotopy zero curve for lamella dome problem (Gill-Murray preconditioner).

n	AGH:2	AGH:4	AGH:6	AGS:2	AGS:4	AGS:6	GS	FGS	Q
21	22,452,1 $(4 \rightarrow 10)$	19,536,1 $(4 \rightarrow 12)$	16,528,1 $(4 \rightarrow 10)$	23,508,1 $(4 \rightarrow 10)$	18,387,1 $(4 \rightarrow 12)$	*	*	*	*
69	38,1374,1 $(8 \rightarrow 18)$	34,1347,1 $(8 \rightarrow 18)$	35,1576,1 $(8 \rightarrow 18)$	40,1331,1 $(8 \rightarrow 18)$	39,1442,1 $(8 \rightarrow 18)$	*	*	(4) * (9)3,75,1	*

restart cycles and was not small, (4) stagnation occurred for GMRES-like methods. In Tables 1–3, the numbers in parentheses (k) show the restart values k whenever applicable. For AGS and AGH, the notation $(a \rightarrow b)$ shows the initial restart value

TABLE 4
Iterative solver execution time in seconds for circuit problems (ILU preconditioner).

n	AGH:2	AGH:4	AGH:6	AGS:2	AGS:4	AGS:6	GS	FGS
31	0.66	0.55	0.55	0.40	0.36	0.36	0.18	1.53
59	0.50	0.45	0.46	0.33	0.31	*	0.08	*
67	3.70	3.53	2.96	1.97	2.09	1.82	0.55	11.04
125	1.51	1.34	1.18	0.98	0.81	0.74	0.56	14.56
468	11.45	11.46	11.45	5.39	5.39	5.38	5.32	*
1854	383.60	386.29	338.76	133.17	133.04	123.31	32.98	*

TABLE 5
Iterative solver execution time in seconds for thin shell problem (Gill-Murray preconditioner).

n	AGH:2	AGH:4	AGH:6	AGS:2	AGS:4	AGS:6	GS
55	25.82	19.68	14.53	*	*	*	*
119	2.70	2.71	2.69	2.43	2.42	2.41	2.44
1239	111.40	107.12	100.90	*	*	*	*

TABLE 6
Iterative solver execution time in seconds for lamella dome problem (Gill-Murray preconditioner).

n	AGH:2	AGH:4	AGH:6	AGS:2	AGS:4	AGS:6	FGS
21	2.24	1.89	1.66	1.74	1.39	*	*
69	29.45	25.93	27.27	24.26	23.64	*	38.86

a and the maximum restart value b reached by the adaptive strategy anywhere along the homotopy zero curve.

The initial k values for the problems are chosen to compare GMRES-like methods when: (1) GMRES(k) does not exhibit near stagnation behavior at all ($n = 468$ circuit problem, $n = 119$ shell problem), (2) near stagnation causes an increase in k for *all* the matrices along the curve (circuit problem $n = 125$), and (3) near stagnation is detected for *some* matrices along the curve. In the first case, adaptive and nonadaptive GMRES(k) perform the same. In the second and third cases, GS with k the same as the initial k in AGMRES(k) reaches no final solution, which has a disastrous effect on the outer (client) algorithm calling GS. An adaptive strategy is especially useful when the degree of difficulty of linear systems in a sequence varies, and it is hard to predict what value of k is needed for convergence of GMRES(k) on all the linear systems. Even for a small shell problem ($n = 55$), k can vary between 8 and 20, and the number of iterations can vary between a low of 1 and a high of 1239. The AGH:4 adaptive strategy on one of the matrices from the "vref" circuit is shown in Fig. 8, where a circle indicates an increase in the subspace dimension ($k = 8$ initially). The adaptive strategy is invoked for the first time rather early—to jump off the near-stagnation "plateau" after the 24th iteration. The subspace dimension is increased once more (after 172 iterations), which suggests that AGH:4 has enough vectors to solve the system within the iteration limit. Note that a smaller iteration limit would cause AGH:4 to increase the subspace dimension quicker.

For every circuit problem, GS with k the maximum restart value produced by the adaptive strategy converges faster than AGS or AGH. However, neither AGS nor GS always converges for the shell problems (Table 2) with $n = 55$, 1239, due to the failure of the modified Gram-Schmidt process to accurately orthogonalize the Krylov subspace basis. When the Krylov subspace basis vectors become nearly linearly dependent, the

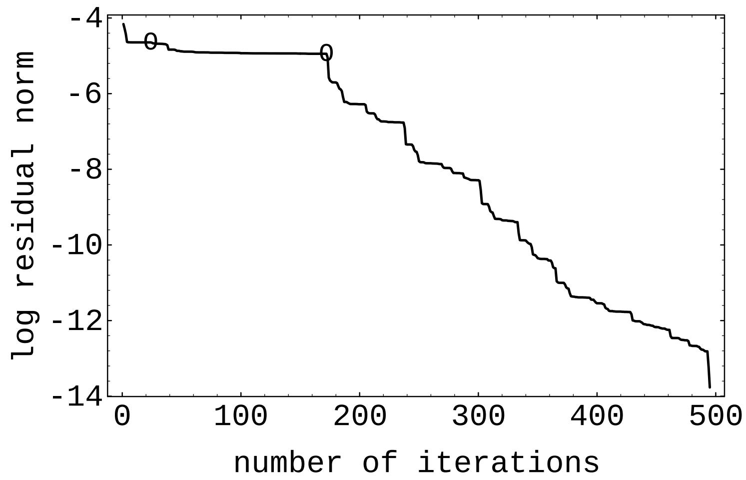

FIG. 8. *Effect of adaptive strategy on residual norm.*

orthogonalization error of the modified Gram-Schmidt process grows with the increase in the number of Krylov subspace basis vectors to be orthogonalized. For example, AGS:6 fails on all lamella dome problems whereas AGS:2 and AGS:4 succeed (Table 3). The same situation is encountered for the circuit problem with $n = 59$. All these failures just mentioned ensue from the estimated condition number of the GMRES least squares problem indicating numerical singularity. AGH successfully passes the condition number test on all the problems, producing a more accurate orthogonal basis for the Krylov subspace, but also spending more CPU time. For example, good candidates for a CPU time comparison (Tables 1, 4) are AGH:4, AGS:4 for the $n = 31$ circuit problem and AGH:2, AGS:2 for the $n = 59$ circuit problem, since they exhibit similar adaptive behavior in terms of iteration counts. In both cases, AGS outperforms AGH only by a factor of 1.5. This factor varies slightly depending on the performance of the adaptive strategy and the orthogonalization accuracy for a particular problem. If the maximum number of AGS iterations is large relative to n (e.g., dome problem with $n = 69$), then AGH is only a little slower than AGS (Table 6), but for large n ($n = 1854$ circuit problem), AGH is slower than AGS by a factor of 2.8 and also takes more iterations (Tables 1, 4). When a high accuracy solution is required, it is hard to predict which orthogonalization procedure will lead to the smallest number of iterations. For example, from Table 1, AGH produces a lower maximum number of iterations than AGS for some cases but not all cases.

The number of restart cycles in AGMRES(k) depends on the increment value m. Tables 1–3 indicate that even a small increment in the restart value leads to convergence. However, if the increment value is too small, often extra restart cycles are executed, which increases the average number of iterations in all cases, comparing $m = 2$ and $m = 6$. As m grows, the execution time of smaller test problems decreases since the cost of an extra iteration in a restart cycle is essentially the same as the cost of earlier ones and the later iterations sometimes contribute the most towards convergence as noted in [18]. For large problems ($n = 1854$), the cost of the later added iterations becomes significant and increases CPU time (compare AGH:2 and AGH:4 for $n = 1854$ in Table 4). Thus, some moderate value for m seems best for the proposed adaptive strategy.

It is clear from the data presented that AGMRES(k) outperforms both flexible GMRES($\hat{k}$) and QMR. Since restarted flexible GMRES($\hat{k}$) requires twice as much memory as restarted AGMRES(k), the restart value for FGS is taken as $\hat{k} = k/2$.

Whenever FGS converges, it requires more work per iteration than AGH because a new preconditioner is computed at each FGS iteration. The QMR algorithm, subroutine DUQMR from QMRPACK [5], fails to find a solution with the required accuracy for any of the problems, but can solve some linear systems along the homotopy zero curve if the error tolerance is relaxed.

5. Conclusions. For applications requiring high accuracy linear system solutions, the adaptive GMRES(k) algorithm proposed here, which is based on Householder transformations and monitoring the GMRES convergence rate, outperforms several standard GMRES variants and QMR. The superiority is not in CPU time or memory requirements, but in robustness as a linear system solution "server" for a much larger nonlinear analysis "client" computation. The extra cost of Householder transformations is well justified by the improved reliability, and the adaptive strategy is well suited to the need to solve a sequence of linear systems of widely varying difficulty. In the context of large scale, multidisciplinary, nonlinear analysis it is hard to imagine not wanting the various components to be adaptive.

The high accuracy requirement causes types of failures in GMRES not usually seen in moderate accuracy uses of GMRES, and monitoring condition number estimates and the onset of stagnation within GMRES becomes crucial. Furthermore, "sanity" checks (e.g., residual norm should be nonincreasing) within a GMRES algorithm must be modified for high accuracy. For instance, the limit on the number of iterations must be increased, and depending on other factors, an increasing residual norm may or may not dictate an abort (such details are in the pseudocode in §2). Finally, the three test problems are rather different, so the details of AGMRES(k) have not been tuned to one particular class of problems.

Future work should investigate a strategy for both increasing and decreasing the Krylov subspace dimension depending on measures of progress. Ideally the accuracy requested of the linear solver should match the accuracy actually required by the calling nonlinear analysis program, although in large scale computation it may be extremely difficult to estimate this minimum required accuracy.

Acknowledgements. The authors gratefully acknowledge very helpful conversations with William McQuain, Robert Melville, and Homer Walker.

REFERENCES

[1] C. H. Bischof and P. T. P. Tang, *Robust incremental condition estimation*, Tech. Rep. CS-91-133, LAPACK Working Note 33, Computer Sci. Dept., Univ. of Tennessee, May, 1991.

[2] Å. Björck, *Solving linear least squares problems by Gram-Schmidt orthogonalization*, BIT, 7 (1967), pp. 1–21.

[3] P. N. Brown and H. F. Walker, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., to appear.

[4] C. DeSa, K. M. Irani, C. J. Ribbens, L. T. Watson, and H. F. Walker, *Preconditioned iterative methods for homotopy curve tracking*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 30–46.

[5] R. W. Freund and N. M. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, part II*, Tech. Rep. 90.46, RIACS, NASA Ames Research Center, November, 1990.

[6] R. W. Freund and N. M. Nachtigal, *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[7] P. E. Gill and W. Murray, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Programming, 7 (1974), pp. 311–350.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 2nd ed., 1989.

[9] S. M. Holzer, R. H. Plaut, A. E. Somers, Jr., and W. S. White, *Stability of lattice structures under combined loads*, J. Engrg. Mech., 106 (1980), pp. 289–305.

[10] K. M. Irani, M. P. Kamat, C. J. Ribbens, H. F. Walker, and L. T. Watson, *Experiments with conjugate gradient algorithms for homotopy curve tracking*, SIAM J. Optim., 1 (1991), pp. 222–251.

[11] R. K. Kapania and T. Y. Yang, *Formulation of an imperfect quadrilateral doubly curved shell element for postbuckling analysis*, AIAA J., 24 (1986), pp. 310–311.

[12] W. D. McQuain, R. C. Melville, C. J. Ribbens, and L. T. Watson, *Preconditioned iterative methods for sparse linear algebra problems arising in circuit simulation*, Comput. Math. Appl., 27 (1994), pp. 25–45.

[13] R. C. Melville, S. Moinian, P. Feldmann, and L. T. Watson, *Sframe: an efficient system for detailed DC simulation of bipolar analog integrated circuits using continuation methods*, Analog Integrated Circuits Signal Processing, 3 (1993), pp. 163–180.

[14] R. C. Melville, Lj. Trajković, S.-C. Fang, and L. T. Watson, *Artificial parameter homotopy methods for the DC operating point problem*, IEEE Trans. Computer-Aided Design, 12 (1993), pp. 861–877.

[15] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[16] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual method for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856 – 869.

[17] H. A. van der Vorst and C. Vuik, *GMRESR: a family of nested GMRES methods*, Numer. Linear Algebra Appl., 1 (1994), pp. 369–386.

[18] H. A. van der Vorst and C. Vuik, *The superlinear convergence behaviour of GMRES*, J. Comp. Appl. Math., 48 (1993) pp. 327–341.

[19] H. F. Walker, *Implementation of the GMRES method using Householder Transformations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 152–163.

[20] H. F. Walker, private communication, 1996.

[21] L. T. Watson, *Globally convergent homotopy methods: a tutorial*, Appl. Math. Comput., 31BK (1989), pp. 369–396.

[22] L. T. Watson, S. M. Holzer, M. C. Hansen, *Tracking nonlinear equilibrium paths by a homotopy method*, Nonlinear Anal., 7 (1983), pp. 1271–1282.

[23] L. T. Watson, S. C. Billups, and A. P. Morgan, *Algorithm 652: HOMPACK: A suite of codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software., 13 (1987), pp. 281–310.

[24] T. Y. Yang, R. K. Kapania, and S. Saigal, *Accurate rigid-body modes representation for a nonlinear curved thin-shell element*, AIAA J., 27 (1989), pp. 211–218.