The Subspace Iteration Method in Protein Normal Mode Analysis

REZA SHARIFI SEDEH,[1] MARK BATHE,[2] KLAUS-JÜRGEN BATHE[1]
[1]*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139*
[2]*Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139*

Received 10 September 2008; Revised 14 January 2009; Accepted 19 January 2009
DOI 10.1002/jcc.21250
Published online 30 April 2009 in Wiley InterScience (www.interscience.wiley.com).

Abstract: Normal mode analysis plays an important role in relating the conformational dynamics of proteins to their biological function. The subspace iteration method is a numerical procedure for normal mode analysis that has enjoyed widespread success in the structural mechanics community due to its numerical stability and computational efficiency in calculating the lowest normal modes of large systems. Here, we apply the subspace iteration method to proteins to demonstrate its advantageous properties in this area of computational protein science. An effective algorithm for choosing the number of iteration vectors in the method is established, offering a considerable improvement over the original implementation. In the present application, computational time scales linearly with the number of normal modes computed. Additionally, the method lends itself naturally to normal mode analyses of multiple neighboring macromolecular conformations, as demonstrated in a conformational change pathway analysis of adenylate kinase. These properties, together with its computational robustness and intrinsic scalability to multiple processors, render the subspace iteration method an effective and reliable computational approach to protein normal mode analysis.

 J Comput Chem 31: 66–74, 2010

Key words: frequency; mode shape; macromolecule; conformational change pathway

Introduction

Normal mode analysis (NMA) plays an important role in relating the conformational dynamics of proteins to their biological function.[1] In classical NMA,[2,3] protein atomic degrees of freedom are treated explicitly in solving the generalized eigenvalue problem in a biologically relevant conformation, typically for the lowest twenty to one hundred normal modes that represent the largest conformational fluctuations of the molecule. In the analysis of conformational transitions, numerous normal mode analyses may be performed for the same protein in nearby conformations.[4]

NMA provides a considerable computational advantage over molecular dynamics because of the elimination of time-integration and explicit solvent degrees of freedom. Nevertheless, significant effort has been directed towards further improving the computational efficiency of NMA to enable its application to ever-larger supramolecular complexes including viral capsids, molecular motors, and the ribosome (ref. 5 and references therein). Particular attention has been directed to the development and application of coarse-grained protein models such as elastic network and related models,[6,7] whereas somewhat less attention has been paid to the development of algorithms that improve the computational efficiency of all-atom protein NMA itself. Such developments are of interest because they preserve the explicit representation of atomic degrees of freedom and their solvent-mediated interactions as modeled by implicit solvent force-fields. The explicit representation of atomic interactions is important to model accurately a number of biological processes, including interactions between proteins and nucleic acids,[8] as well as small molecules in rational drug design.[9] Additionally, the role of allosteric regulation of binding affinity and catalysis by at-a-distance mutations remains an interesting and open area of research that may require all-atom modeling to understand fully.[10]

The subspace iteration method was originally developed for the solution of frequencies and mode shapes of macroscopic structures such as buildings and bridges using finite element analysis (FEA).[11,12] In those applications, relatively few frequencies and corresponding mode shapes were sought, such as the lowest 10–20 eigenpairs in models containing a total of 1000–

Additional Supporting Information may be found in the online version of this article.

Correspondence to: M. Bathe; e-mail: mark.bathe@mit.edu or: K.-J. Bathe; e-mail: kjb@mit.edu

10,000 degrees of freedom. Since its development, however, the subspace iteration method has been used extensively in the FEA of considerably larger systems reaching millions of degrees of freedom, and naturally has attracted significant attention for improvements as a result (see for example refs. 13–20).

The subspace iteration method is a particularly attractive approach to protein NMA because the procedure (1) is designed specifically for the calculation of the lowest eigenpairs of large systems; (2) uses previously calculated eigenvectors from nearby conformations to speed up significantly the solution of eigenpairs in nearby conformations of interest; (3) is computationally robust; and (4) is amenable to parallel-processing.

The original development of the method was based on the earlier use of the Ritz method, and relates to the works of Bauer[21] and Rutishauser.[22] Key developments for its practical use in structural engineering were the specific steps in the iteration method, the construction of the starting iteration vectors, the use of an effective number of iteration vectors, the use of error measures, and the Sturm sequence check.[11] A convergence analysis of the subspace iteration method is given in ref. 23. The method is also abundantly used in the solution of linearized buckling problems,[24] which is applicable to calculations of the stability of the cytoskeletal polymers filamentous actin and microtubules, as well as viral capsids and other supramolecular assemblies with mechanically related biological function.[7]

An additional leading approach to NMA in the structural mechanics community is the Lanczos method,[25] advanced particularly by Paige [26] and others.[27] Initially, the Lanczos method exhibited instabilities due to loss of orthogonality of the iteration vectors employed. This shortcoming, however, has been largely overcome, and when implemented properly the method is highly efficient. A particular asset of the method is that computational effort scales about linearly (neglecting the effort for the initial factorization) with the number of eigenpairs sought, a property that is not generally satisfied by the traditional subspace iteration method. An important property of both the subspace iteration and Lanczos procedures is that they solve directly for the eigenpairs sought instead of calculating intermediate matrices first, as if all eigenvalues were desired. This property contrasts with the approach of the Householder–QR method,[24] for example, which becomes prohibitively expensive computationally and in memory as the size of coefficient matrices increases. At present, the Lanczos and subspace iteration methods are the two most widely used techniques for the solution of large eigenvalue problems in FEA, when coefficient matrices are of order 10,000–10,000,000. For these reasons, any significant improvements to these methods are of great interest.

Recently, considerable effort has been directed towards using parallel processing in FEA, in shared-memory and distributed-memory processing modes. Whereas the Lanczos method can intrinsically (largely) be parallelized only in the factorization of the stiffness matrix and the forward reduction and back-substitution of the *individual* vectors, the subspace iteration method allows in addition the parallel solution of *multiple* iteration vectors which can result in a large computational benefit. However, there is also interest in improving the method in other ways, and in particular, for the solution of eigenproblems in which relatively many eigenpairs need to be calculated.

As mentioned earlier, a key step in the subspace iteration method is the establishment of effective starting iteration vectors, which implies using an optimal number of iteration vectors. The objective of the present work is to apply the subspace iteration method to the normal mode analysis of proteins, and to introduce a significant improvement upon the original implementation regarding the choice of the number of iteration vectors. In the following sections, we first review briefly the standard subspace iteration method and discuss its inherent value for the solution of frequencies and mode shapes of proteins. We, subsequently, present a new algorithm to establish an effective number of iteration vectors, illustrating the use of this algorithm in some applications. A particularly important observation is that computational effort increases linearly with the number of eigenpairs sought in the solutions obtained with the improved subspace iteration method, as in the Lanczos method. To focus on our new development only, and to compare results obtained with the traditional and improved methods, we employ a basic implementation without parallelization of the code, running in-core on a single processor workstation. Moreover, we provide only relative solution times, which are largely independent of the machine used. Although these times thereby represent practically "machine-independent" algorithmic improvements, actual solution times will naturally depend on the specific machine employed and will decrease as computational hardware becomes more efficient.

The Basic Subspace Iteration Method

We consider the generalized eigenvalue problem

$$\mathbf{K}\boldsymbol{\varphi} = \lambda \mathbf{M}\boldsymbol{\varphi} \tag{1}$$

where $\mathbf{K}$ and $\mathbf{M}$ are symmetric matrices of order n, $\mathbf{K}$ is positive definite, and $\mathbf{M}$ is positive semidefinite. We seek the smallest p eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ and corresponding eigenvectors $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_p$ with the ordering

$$\lambda_1 \leq \lambda_2 \ldots \leq \lambda_p. \tag{2}$$

The eigenpairs $(\lambda_i, \boldsymbol{\varphi}_i)$ satisfy

$$\mathbf{K}\boldsymbol{\varphi}_i = \lambda_i \mathbf{M}\boldsymbol{\varphi}_i; \quad i = 1, \ldots, p \tag{3}$$

and

$$\begin{aligned} \boldsymbol{\varphi}_i^T \mathbf{M}\boldsymbol{\varphi}_j &= \delta_{ij} \\ \boldsymbol{\varphi}_i^T \mathbf{K}\boldsymbol{\varphi}_j &= \lambda_i \delta_{ij} \end{aligned} \tag{4}$$

where δ_{ij} is the Kronecker delta. The basic equations used in the subspace iteration method are as follows[24]:

Step 1: Establish q starting iteration vectors in $\mathbf{X}_1$

Step 2: Iterate with $k = 1, 2, 3, \ldots$, until convergence

$$\mathbf{K}\overline{\mathbf{X}}_{k+1} = \mathbf{M}\mathbf{X}_k \tag{5}$$

$$\begin{aligned} \mathbf{K}_{k+1} &= \overline{\mathbf{X}}_{k+1}^T \mathbf{K}\overline{\mathbf{X}}_{k+1} \\ \mathbf{M}_{k+1} &= \overline{\mathbf{X}}_{k+1}^T \mathbf{M}\overline{\mathbf{X}}_{k+1} \end{aligned} \tag{6}$$

$$\mathbf{K}_{k+1}\mathbf{Q}_{k+1} = \mathbf{M}_{k+1}\mathbf{Q}_{k+1}\mathbf{\Lambda}_{k+1} \tag{7}$$

$$\mathbf{X}_{k+1} = \overline{\mathbf{X}}_{k+1}\mathbf{Q}_{k+1} \tag{8}$$

Step 3: Perform the Sturm sequence check.

Hence, the procedure consists of three distinct solution steps. First, the q starting iteration vectors in $\mathbf{X}_1$ are established, $q > p$, where $\mathbf{X}_1$ is a matrix of dimension $n \times q$. Second, iteration is performed using eqs. (5)–(8), for $k = 1, 2, \ldots$ until the convergence tolerance below is satisfied, where $\mathbf{Q}_{k+1}$ and $\mathbf{\Lambda}_{k+1}$ store the eigenvectors and eigenvalues corresponding to the subspace matrices $\mathbf{K}_{k+1}$ and $\mathbf{M}_{k+1}$. Finally, the Sturm sequence check is performed.

Let $\lambda_i^{(k)}$ be the approximation for λ_i calculated in the $(k-1)^{\text{th}}$ iteration, we have convergence to an accuracy of $2 \times s$ digits in the eigenvalues when for $i = 1, \ldots, p$ (see ref. 24)

$$\left[1 - \frac{\left(\lambda_i^{(k)}\right)^2}{\left(\mathbf{q}_i^{(k)}\right)^T \mathbf{q}_i^{(k)}}\right]^{1/2} \leq 10^{-2s} \tag{9}$$

where $\mathbf{q}_i^{(k)}$ is the vector in the matrix $\mathbf{Q}_k$ corresponding to $\lambda_i^{(k)}$. The eigenvectors will only be accurate to s digits and the theoretical convergence rate of the vectors is λ_i/λ_{q+1}. Thus, there is a higher convergence rate for a smaller eigenvalue and its corresponding eigenvector. Although these convergence rates correspond to the theoretical values,[23,24] they are usually also observed in actual computations. The Sturm sequence check is carried out to ensure that the lowest p eigenpairs, that is, (λ_i, $\boldsymbol{\varphi}_i$), $i = 1, \ldots, p$, have indeed been calculated.[11,24] If the Sturm sequence check is not passed, the iteration is continued with a larger number of iteration vectors.

Considering eqs. (5)–(8), it is seen that the method can be programmed efficiently for parallel computations. The factorization of the coefficient matrix and the forward reductions and back-substitutions of each individual vector can be parallelized. In addition, the solution of the q vectors can be distributed to different processors and also the computation of the subspace matrices $\mathbf{K}_{k+1}$ and $\mathbf{M}_{k+1}$ can be parallelized.

An important difference between the coefficient matrices of structural FE assemblages and of proteins is that the latter have much larger bandwidths because of long-range nonbonded electrostatic, and to a lesser extent van der Waals, interactions that introduce broad coupling between protein atoms. Thus, for a given number of degrees of freedom, the factorization of the matrix and solution of the vectors in eq. (5) constitute a much larger computational effort than in standard FE solutions. Although parallel processing can be very important for this reason, we do not address this computational issue further in the present work.

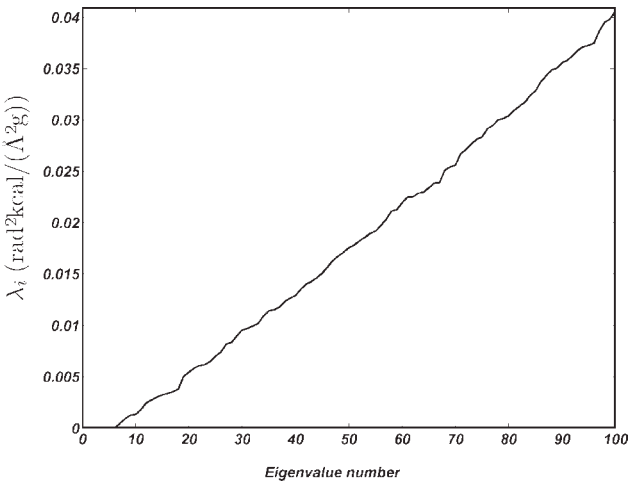

Figure 1. The lowest one hundred eigenvalues (λ_i) of T4-lysozyme (Protein Data Bank ID 3LZM).[30] (The first six zero eigenvalues correspond to rigid body modes.)

Using the earlier equations, it is critical to establish effective starting iteration vectors for two reasons. First, if the subspace of these vectors contains the exact eigenvectors, theory states that a single iteration will result in the exact eigenvalues and vectors sought. Here, we simply use the algorithm of ref. 11 (also given in ref. 24), to construct the starting iteration vectors. In cases where better starting vectors are known from an existing solution, such as in conformational change pathway analyses of proteins where eigensolutions may be performed numerous times for small changes in protein conformation,[4] the algorithm of ref. 24 is used only for the first eigensolution. Thereafter, the previous solution from the nearest-neighbor conformation provides the starting iteration vectors for the next eigensolution. Second, an effective value of q needs to be used because the convergence rate to an eigenvector is given by λ_i/λ_{q+1}. If q ($>p$) is small, a relatively large number of iterations are required to converge. In contrast, if q is large, fewer iterations are required for convergence, but each iteration is computationally more costly. Thus, use of an optimal value of q is highly desirable. Calculation of an effective value of q for the frequency and mode shape solutions of proteins is addressed in the next section.

The Algorithm to Calculate the Number of Starting Iteration Vectors

An important observation regarding proteins is that the magnitudes of their eigenvalues increase nearly linearly with increasing wave-number,[28,29] as shown for T4-lysozyme in Figure 1. This characteristic of proteins may be used to find an effective value of q for the subspace iteration method.

Assume that we order the iteration vectors in $\mathbf{X}_k$ naturally so that they correspond to increasing eigenvalues, with the first vector corresponding to λ_1. Then the last iteration vector to converge is the p^{th} vector in $\mathbf{X}_k$ and its rate of convergence is λ_p/λ_{q+1}. Additionally, after the i^{th} iteration, the norm of the vector

difference between the p^{th} **M**-orthonormalized eigenvector and its current approximation (the 'error vector $\boldsymbol{\varepsilon}$') is given by

$$||\boldsymbol{\varepsilon}(\text{current})|| = \left(\lambda_p \Big/ \lambda_{q+1}\right)^i ||\boldsymbol{\varepsilon}(\text{initial})|| \tag{10}$$

where $\|\boldsymbol{\varepsilon}(\text{initial})\|$ is the initial error vector. To reach s-digits of accuracy in the eigenvector we need

$$\left(\lambda_p \Big/ \lambda_{q+1}\right)^i ||\boldsymbol{\varepsilon}(\text{initial})|| \leq 10^{-s} \tag{11}$$

and, therefore, require l iterations for the vector to converge, where l is given by

$$l = \frac{\ln\left(10^{-s} \Big/ ||\boldsymbol{\varepsilon}(\text{initial})||\right)}{\ln\left(\lambda_p \Big/ \lambda_{q+1}\right)}. \tag{12}$$

Next, we use the fact that the eigenvalue magnitudes increase linearly and assume that for different values of q, the norm of the initial error vector for the p^{th} iteration vector is the same. Additionally, the first six eigenvalues are zero. This implies that the **K** matrix is singular. To use the subspace iteration method, we perform a shift ρ on the **K** matrix to have a positive definite matrix, see ref. 24. We use ρ to be a very small value, $\rho = -1\text{E-}6$. Therefore, λ_p/λ_{q+1} is approximately equal to $(p - 6 - \rho)/(q - 5 - \rho)$. Since ρ is very small, it can be neglected and λ_p/λ_{q+1} is approximated as $(p - 6)/(q - 5)$. Then eq. (12) gives us directly

$$l = \frac{\ln\left(10^{-s} \Big/ ||\boldsymbol{\varepsilon}(\text{initial})||\right)}{\ln\left((p-6) \Big/ (q-5)\right)} \tag{13}$$

However, an operation count tells that the following number of numerical operations are needed for l iterations with q vectors[24]

$$TCC = \frac{\ln\left(10^{-s} \Big/ ||\boldsymbol{\varepsilon}(\text{initial})||\right)}{\ln\left((p-6) \Big/ (q-5)\right)} \left(2nmq + 2nq^2 + 3nq\right) \tag{14}$$

where TCC is the Total Cost of Computation for l iterations, n is the order of the **K** and **M** matrices, and m is the half-bandwidth (assumed to be full) of the **K** matrix. As the column heights of **K** vary, an average or effective value for m must be used.[24] Although we refer to TCC in eq. (14), in reality we only have the total number of *arithmetical* operations. As our only purpose is to find an effective value of q for each p, and we also know that

$$c = \ln\left(10^{-s} \Big/ ||\boldsymbol{\varepsilon}(\text{initial})||\right)$$

where c is an unknown constant, we may use

$$\text{TCC} = \frac{c}{\ln\left((p-6) \Big/ (q-5)\right)} \left(2nmq + 2nq^2 + 3nq\right). \tag{15}$$

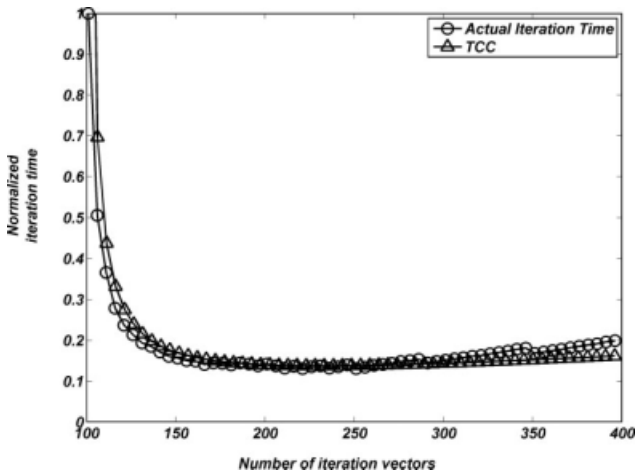

Figure 2. Normalized actual iteration time and normalized TCC to calculate the first one hundred eigenvalues for T4-lysozyme (Protein Data Bank ID code 3LZM).[30]

Minimizing this expression with respect to q we find an approximation for the best q to obtain the p eigenvalues and vectors in the least amount of computational time. Because a closed-form solution does not exist, we solve for q by iteration. Note that this analysis does not provide the actual computational effort required (since the constant c is unknown) but only that the minimum is obtained when using the value of q given by minimizing TCC in eq. (15).

Figure 2 shows the normalized actual solution time and TCC to calculate the lowest 100 eigenvalues with six digits of accuracy for T4-lysozyme using different numbers of iteration vectors. The iteration times are normalized by the maximum actual iteration time and, since the constant c in eq. (15) is unknown, TCC is scaled such that the iteration times are equal at the minimum of TCC.

As seen in Figure 2, prediction of the relative computational cost of calculating the lowest eigenvalues with different numbers of iteration vectors by eq. (15) is acceptable. Next we illustrate the use of the value of q in the normal mode analyses of two proteins.

Illustrative Solutions

In this section we use the subspace iteration method for the calculation of the frequencies and normal modes of two proteins. In each case we use the standard subspace iteration method as published in refs. 11, 24 including the algorithm to construct *all* starting iteration vectors. We use the standard value $q = \min\{2p, p + 8\}$, referred to as the "traditional subspace iteration method," and this method with the value of q that minimizes TCC in eq. (15), referred to as the "improved subspace iteration method." We intentionally do not use any other acceleration techniques, such as given for example in ref. 13, to identify clearly the improvements achieved solely by use of the value of q derived earlier.

In each solution we employ the skyline solver of ref. 24 for eq. (5). Although we recognize that a sparse solver could lead to

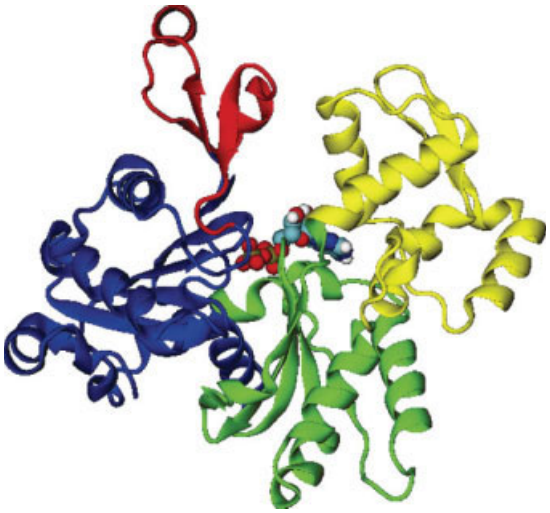

Figure 3. G-actin-ADP. Schematic representation of the energy-minimized molecular structure analyzed with subdomains colored according to the definition of Kabsch et al.,[35] Subdomain 1 is colored blue, subdomain 2 is colored red, subdomain 3 is colored green, and subdomain 4 is colored yellow. ADP is shown in van der Waals representation. Figure rendered using Visual Molecular Dynamics.[36]

significantly improved solution times,[31] we do not expect our fundamental observations regarding the performance of the method to be affected. We note that the solution times given always include all operations of the subspace iterations. Additionally, in an effort to present machine-independent conclusions regarding performance of the algorithms, we present normalized solution times instead of actual solution times, where normalized time is equal to actual time divided by the maximum solution time measured in each case.

G-actin

The initial structure of ADP-bound G-actin is taken from the work of Otterbein et al.[32] (Protein Data Bank ID 1J6Z; residue numbers 4–372). The stiffness matrix of order 10,608 for this protein was computed in CHARMM version 34b1[33] using the implicit solvation model EEF1.[34] Steepest descent minimization followed by Adopted-Basis Newton–Raphson minimization is performed in the presence of successively reduced harmonic constraints on backbone atoms to achieve a final root-mean-square (RMS) energy gradient of 2×10^{-4} kcal/(mol $\times$ Å) with corresponding RMS deviation between the X-ray and energy-minimized structures of 1.4 Å (Fig. 3). Computations are performed on an Intel Xeon 5120 with 1.86 GHz and 4 GB RAM in single processor mode.

Considering the eigenvalue problem, different numbers of the lowest eigenvalues with six digits of accuracy of this protein have been obtained using the traditional and improved subspace iteration methods. Figure 4 provides normalized solution times versus the required number of lowest eigenvalues for G-actin, and also provides in parentheses the number of iteration vectors

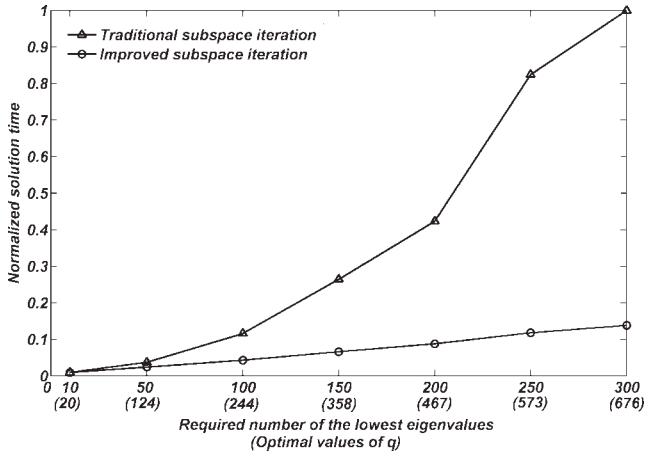

Figure 4. Normalized solution times versus required number of the lowest eigenvalues with six digits of accuracy for G-actin (Protein Data Bank ID 1J6Z)[32] using the traditional and improved subspace iteration methods; the value of q used in each case with the improved subspace iteration method is given in parentheses.

q used in the improved subspace iteration method in each case. It is evident that a significant improvement in the subspace iteration method is achieved by use of the calculated values of q.

As already noted, normalized solution times in Figure 4 are defined as the actual solution times divided by the maximum solution time encountered in the analysis. The maximum solution time (13,939 seconds clock-time) in this case is the time required to compute the lowest 300 eigenpairs with the traditional subspace iteration method. This solution time is quite large for the reasons mentioned earlier.

Pertussis Toxin

The next protein examined is Pertussis Toxin (chains A–F). Initial coordinates are taken from the work of Stein et al.[37] (Protein

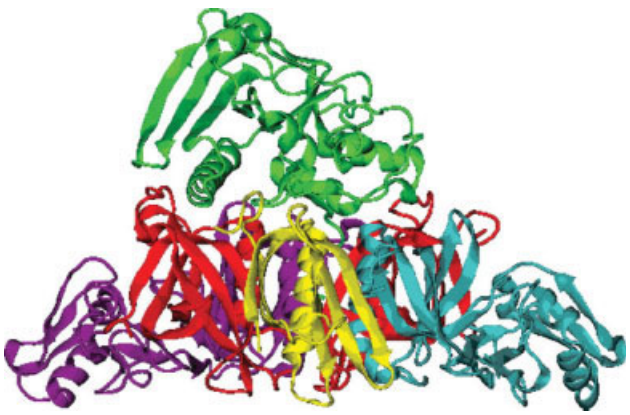

Figure 5. Pertussis toxin. Schematic representation of the energy-minimized molecular structure analyzed with subdomains colored according to the definition of Stein et al.,[37] S1 is colored green, S2 is cyan, S3 is purple, S4 is red, and S5 is yellow. Figure rendered using Visual Molecular Dynamics.[36]

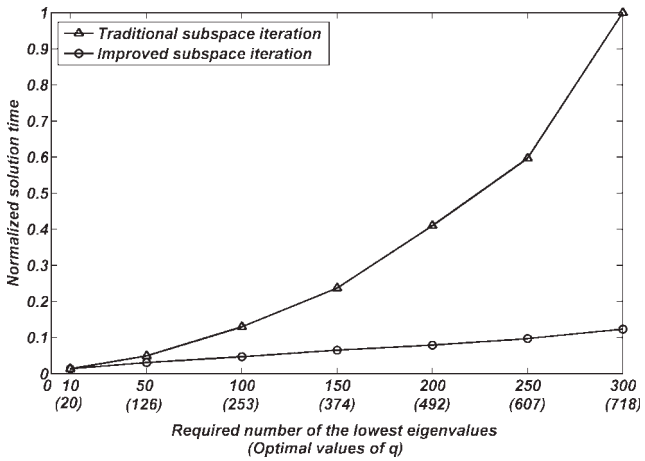

Figure 6. Normalized solution times versus required number of the lowest eigenvalues with six digits of accuracy for one of two molecules from pertussis toxin (Protein Data Bank ID 1PRT; Chains A-F) [37] using the traditional and improved subspace iteration methods; the value of q used in each case with the improved subspace iteration method is given in parentheses.

Data Bank ID 1PRT). Like for G-actin, CHARMM version 34b1[33] with the implicit solvation model EEF1[34] is used to obtain the energy minimized structure (Fig. 5) and calculate the Hessian, which has dimension of order 26,664. Steepest descent minimization followed by adopted-basis Newton–Raphson minimization is performed in the presence of successively reduced harmonic constraints on backbone atoms to achieve a final root mean square (RMS) energy gradient of 3×10^{-4} kcal/(mol Å) with corresponding RMS deviation between the X-ray and energy-minimized structures of 1.6 Å. Computations are also performed on an Intel Xeon 5120 with 1.86 GHz and 4 GB RAM in single processor mode.

Figure 6 shows the measured normalized solution times versus the required number of the lowest eigenvalues for this molecule, and also gives in parentheses the number of iteration vectors q used in the improved subspace iteration method in each case. Again, significant computational savings are achieved when the improved iteration method is used.

Adenylate Kinase

To illustrate the benefit of employing the subspace iteration procedure to analyze conformational change pathways of proteins, we apply the procedure to the open-to-closed transition of adenylate kinase (PDBIDs 4AKE[38] and 1AKE[39] for the open and closed conformers, respectively)(Figs. 7a and 7b). In the absence of molecular dynamics or other all-atom trajectory, we employ the elastic-based FE model applied previously to protein NMA to generate the conformational change pathway.[7] The initial model is defined by the open conformation of the protein. Following ref. 7 the molecular volume is defined by the solvent excluded surface (SES) using MSMS ver. 2.6.1.[40] This SES is then decimated to a coarsened surface using the surface simplification algorithm QSLIM,[41–43] as implemented in MeshLab.[44] Finally, the decimated SES is imported into the finite element analysis program ADINA ver. 8.5 (Watertown, MA), where the molecular volume is meshed automatically using 3D four-node tetrahedral elements.[7] The protein is assumed to behave as a linear, isotropic material with homogeneous mass density of 1420 kg/m^3, elastic Young's modulus of 4.9 GPa, and Poisson ratio of 0.3. The mass density is obtained from the molecular weight and molecular volume of the open conformation. The Young's modulus is obtained by fitting thermal fluctuations of α-carbon atoms in the finite element model to those obtained using the Rotation Translation Block procedure[45,46] at room temperature in CHARMM, where one block per residue and the implicit solvation model EEF1[34] are employed.

The conformational change pathway of adenylate kinase is generated according to the procedure of Tama, Miyashita, and Brooks.[47] Starting from the initial, open conformation, **K** and **M** matrices are generated for the FE model using ADINA. The traditional subspace iteration procedure is then used to calculate the first 100 eigenpairs of the model with four digits of accuracy for the eigenvalues. The FE model interpolation functions are used to interpolate the eigenvectors, $\boldsymbol{\varphi}_i^k$, corresponding to the FE nodal positions to their values, $\mathbf{C}_i^k$, at the positions of the α-carbons, where i and k denote the number of the eigenvector and conformation, respectively. To generate the next conformation, the difference vector between the positions of the α-carbons in the k^{th} conformation and those of the closed conformation, $\Delta\mathbf{r}^k$, is projected onto the eigenvectors corresponding to the α-carbons, $c_i^k = \beta^k\ \Delta\mathbf{r}^k \cdot \mathbf{C}_i^k$, where β^k is a parameter between zero and one[4,47] (Supporting Information). c_i^k is the contribution of the i^{th} eigenvector to the displacement of the α-carbons in the k^{th} step. Positions of all non-α-carbon atoms are updated using the FE displacement interpolation functions in the current conformation. This procedure is repeated until the root-mean-square-difference (RMSD) between the current positions of α-carbons and those of the closed conformer is less than or equal to 1 Å. In this approach to generating the conformational change pathway, the eigenvectors of the current conformation are used as the starting vectors for the eigenvalue problem of the next conformation, excluding the first step, which is also excluded from the solution time per conformation presented below because it constitutes a small and invariant component of the total solution time in each case. An initial conformational change pathway of 1843 conformations is generated, from which subsets of 1001, 101, 11, and 1 conformation are chosen with nearly constant differences in RMSD between α-carbon positions of each successive conformation and the closed conformation (Supporting Information) (Fig. 7c). Computations are performed on an Intel Xeon E5405 with 2.00 GHz and 16 GB RAM in single processor mode.

The solution time per conformation for the subspace iteration procedure decreases monotonically with increasing number of conformations employed in the conformational change pathway (see Fig. 8). Normalized time is equal to the actual solution time divided by the maximum solution time measured in the 100 normal mode case. As an increasing number of conformations is employed, normal mode solutions from neighboring conformations become increasingly better choices for the starting normal

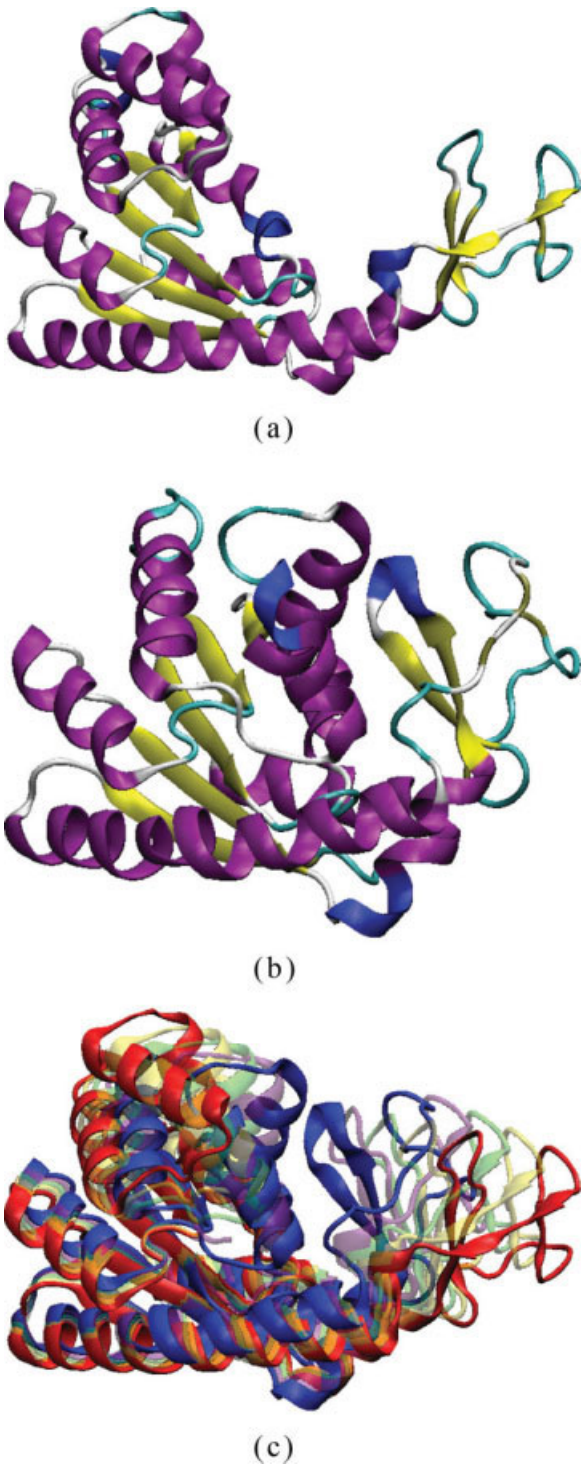

Figure 7. (a) Schematic representation of the open conformation of adenylate kinase (Protein Data Bank ID 4AKE[38]). (b) Schematic representation of the closed conformer of adenylate kinase (Protein Data Bank ID 1AKE).[39] (c) Schematic representation of the open-to-closed transition. The root-mean-square-difference between the positions of α-carbons in the closed conformer and that of the red, yellow, green, violet, and blue conformations is 7.14, 5.25, 3.5, 1.75, and 0 Å, respectively. Figures rendered using Visual Molecular Dynamics.[36]

modes of neighboring conformations, resulting in the observed decrease in solution time per conformation. This result is true whether 20 or 100 eigenvectors are solved for (see Fig. 8), and is additionally expected to be independent of the number of degrees of freedom in the model. Although it is of interest to understand the detailed solution-time properties of the subspace iteration procedure in conformational change pathway analysis (e.g., dependence of solution time per conformation scaling with model size, number of normal modes computed, etc.), such analysis is reserved for future work.

Important Properties of the Subspace Iteration Method

In evaluating the effectiveness of any numerical procedure, it is clearly valuable to make a thorough comparison with existing methods.[2,25,48] In the present case, such comparison is unfortunately complicated by a number of factors, including the requirement that each method employs the same convergence tolerance and is implemented in the optimal manner. Even then, results would depend on whether the computation is performed in- or out-of-core, the type of parallel processing used, the degree of energy-minimization performed in the use of some methods, and so on. While such a comparison would clearly be of value, it is outside the scope of the present work. Nevertheless, we would like to point out several important properties of the subspace iteration procedure, and in particular contrast these properties with corresponding properties of the Lanczos method.

The subspace iteration procedure converges monotonically and robustly to the number of frequencies and mode shapes sought. In each subspace iteration, inverse iteration is performed on a q-dimensional subspace and a Rayleigh–Ritz analysis extracts the "best" approximations to the p normal modes sought. "Best" here refers to minimization of the Rayleigh quo-

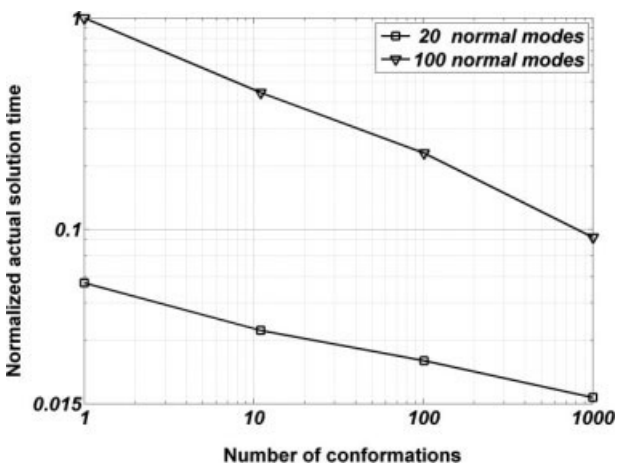

Figure 8. Normalized actual solution time per conformation for the subspace iteration method versus the number of conformations analyzed in the conformational change pathway of adenylate kinase using 100 and 20 normal modes.

tient on the subspace.[23,24] As the q-dimensional subspace is rotated towards the least dominant p-dimensional subspace within each iteration, the NM approximations become increasingly accurate. If only low accuracy in the normal modes is needed, only a few subspace iterations may be required.

Solution time in the Lanczos method scales approximately linearly with the number of eigenpairs computed. The traditional subspace iteration does not typically display this scaling when many frequencies and mode shapes are calculated (e.g., >20) and a single processor is employed. In the present work, however, we observed that the subspace iteration method with the improved selection of the number of iteration vectors also resulted in linear scaling of solution time with the number of normal modes sought. As expected, we additionally observed a significant decrease in computational time when the NMA was performed on multiple neighboring conformations, because the method uses normal mode solutions from neighboring conformations to accelerate subsequent solutions. This is an important property of the subspace iteration procedure that is not a property of methods that start with individual vectors, such as the Lanczos algorithm. Additional acceleration might be achieved for NMA of single conformations by exciting principally the dihedral angles to choose starting vectors that span a subspace that is closer to the required least dominant subspace than the algorithm employed here.[11,24] In addition, acceleration techniques published previously could be implemented.[13,18]

A final important computational property of any NMA procedure is the possibility to use parallel processing (with shared and distributed memory), such as implemented for the Lanczos procedure in the publically available program ARPACK.[49] Although the calculations in the subspace iterations [eqs. (1)–(5)] lend themselves naturally to parallel processing, the actual benefits achievable in comparison to the Lanczos procedure, which operates sequentially on individual vectors, remains to be established. Use of a combination of the basic steps in the subspace iteration and Lanczos methods, using the best ingredients of each technique and taking into account parallel processing, would be of interest to reach a more effective method. Further investigation is required to identify the appropriate next steps to take in this research direction.

Conclusions

The objective of this article was to present the application of the subspace iteration method to the normal mode analysis of proteins and to provide an algorithm for the calculation of an effective number of iteration vectors. We demonstrated use of an algorithm to calculate the number of iteration vectors q to find p eigenpairs that improves the effectiveness of the subspace iteration method significantly for proteins. The algorithm results in computation time scaling linearly with the number of eigenpairs sought, as demonstrated for G-actin and pertussis toxin. The subspace iteration method is well suited to protein NMA because relatively small subsets of the total available normal modes are typically sought and numerous analyses may be performed for relatively similar conformations in conformational change pathway analyses.[4] In such cases, the previously calculated eigensolution provides an excellent set of initial iteration vectors for the subsequent solution, as demonstrated here for the open-to-closed conformational change of adenylate kinase. The subspace iteration method is additionally attractive because it is robust, in that it converges monotonically to the desired eigenvalue solution for any positive semidefinite stiffness matrix. This is of significant utility in all-atom protein NMA for two reasons. First, energy minimization to tight tolerance in the energy gradient is time-consuming and often challenging due to the rugged energy landscape of proteins, and second, energy minimization often distorts the protein structure such that it deviates significantly from the experimental crystal structure. For these reasons, and due to its relative computational efficiency, the robust Rotational Translational Blocks procedure[45,46] has gained significant popularity. However, this procedure assumes single or larger blocks of residues to be rigid, in contrast with the present implementation that retains all atomic degrees of freedom. Although the significant reduction in number of degrees of freedom in the former approach renders its computational efficiency high, an interesting area of future research concerns the integration of computationally robust NMA methods with efficient reduced degree-of-freedom approaches that retain internal residue flexibility, as initially proposed in ref. 45. Incorporation of such procedures into the finite element method would enable simultaneously calculations of protein mechanical response, as well as NMs.

Acknowledgments

Useful discussions with Liliane Mouawad, David Perahia, Daniel ben-Avraham, and Martin Karplus are gratefully acknowledged.

References

1. Cui, Q.; Bahar, I. (editors), Normal Mode Analysis: Theory and Applications to Biological and Chemical Systems; Chapman & Hall/CRC: Boca Raton, 2006.
2. Brooks, B. R.; Janezic, D.; Karplus, M. J Comput Chem 1995, 16, 1522.
3. Brooks, B. R.; Karplus, M. Proc Natl Acad Sci USA 1985, 82, 4995.
4. Tama, F.; Brooks, C. L. In Normal Mode Analysis: Theory and Applications to Biological and Chemical Systems. Cui, Q.; Bahar, I., Eds.; Chapman & Hall/CRC: Boca Raton, 2006; Chapter 6.
5. Tama, F.; Brooks, C. L. Annu Rev Biophys Biomol Struct 2006, 35, 115.
6. Bahar, I.; Atilgan, A. R.; Erman, B. Fold Des 1997, 2, 173.
7. Bathe, M. Proteins Struct Funct Bioinform 2008, 70, 1595.
8. Van Wynsberghe, A.; Li, G. H.; Cui, Q. Biochem 2004, 43, 13083.
9. Sherman, W.; Tidor, B. Chem Biol Drug Des 2008, 71, 387.
10. Lee, J.; Natarajan, M.; Nashine, V. C.; Socolich, M.; Vo, T.; Russ, W. P.; Benkovic, S. J.; Ranganathan, R. Science 2008, 322, 438.
11. Bathe, K. J. Solution methods for large generalized eigenvalue problems in structural engineering, Report UCSESM 71-20, Department of Civil Engineering, University of California, Berkeley, 1971.
12. Bathe, K. J.; Wilson, E. L. Int J Num Meth Eng 1973, 6, 213.

13. Bathe, K. J.; Ramaswamy, S. Computer Meth Appl Mech Eng 1980, 23, 313.
14. Akl, F. A.; Dilger, W. H.; Irons, B. M. Int J Num Meth Eng 1979, 14, 629.
15. Akl, F. A.; Dilger, W. H.; Irons, B. M. Int J Num Meth Eng 1982, 18, 583.
16. Jung, H. J.; Kim, M. C.; Lee, I. W. Comput Struct 1999, 70, 625.
17. Pradlwarter, H. J.; Schueller, G. I.; Szekely, G. S. Comput Struct 2002, 80, 2415.
18. Zhao, Q. C.; Chen, P.; Peng, W. B.; Gong, Y. C.; Yuan, M. W. Comput Struct 2007, 85, 1562.
19. Wang, X.; Zhou, J. Comput Struct 1999, 71, 293.
20. Qlan, Y. Y.; Dhatt, G. Comput Struct 1995, 54, 1127.
21. Bauer, F. L. Zeitschrift für Angewandte Mathematik und Physik 1957, 8, 214.
22. Rutishauser, H. Numer Math 1969, 13, 4.
23. Bathe, K. J. In Formulations and Computational Algorithms in Finite Element Analysis; Bathe, K. J.; Oden, J. T.; Wunderlich, W.; Eds.; Cambridge, MA: MIT Press, 1977; pp. 575–598.
24. Bathe, K. J. Finite Element Procedures; Prentice Hall: New Jersey, 1996.
25. Lanczos, C. J Res Natl Bur Stand 1950, 45, 255.
26. Paige, C. C. IMA J Appl Math 1972, 10, 373.
27. Ericsson, T.; Ruhe, A. Math Comp 1980, 35, 1251.
28. Elber, R.; Karplus, M. Phys Rev Lett 1986, 56, 394.
29. Ben-Avraham, D. Phys Rev B 1993, 47, 14559.
30. Matsumura, M.; Wozniak, J. A.; Sun, D. P.; Matthews, B. W. J Biol Chem 1989, 264, 16059.
31. Bathe, K. J. The Finite Element Method, Chapter in Encyclopedia of Computer Science and Engineering; Wah, B.; Ed.; John Wiley & Sons, 2009; pp. 1253–1264.
32. Otterbein, L. R.; Graceffa, P.; Dominguez, R. Science 2001, 293, 708.
33. Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. J Comput Chem 1983, 4, 187.
34. Lazaridis, T.; Karplus, M. Proteins 1999, 35, 133.
35. Kabsch, W.; Mannherz, H. G.; Suck, D.; Pai, E. F.; Holmes, K. C. Nat 1990, 347, 37.
36. Humphrey, W.; Dalke, A.; Schulten. K. J Mol Graph 1996, 14, 33.
37. Stein, P. E.; Boodhoo, A.; Armstrong, G. D.; Cockle, S. A.; Klein, M. H.; Read, R. J. Struct 1994, 2, 45.
38. Muller, C. W.; Schlauderer, G. J.; Reinstein, J.; Schulz, G. E. Struct 1996, 4, 147.
39. Muller, C. W.; Schulz, G. E. J Mol Biol 1992, 224, 159.
40. Sanner, M. F.; Olson A. J.; Spehner J. C. Biopolymers 1996, 38, 305.
41. Heckbert, P. S.; Garland M. Theory Appl 1999, 14, 49.
42. Garland, M. Quadric-Based Polygonal Surface Simplification; PhD Thesis; Carnegie Mellon University, 1999.
43. Garland, M.; Heckbert P. S. SIGGRAPH 97 1997, 209.
44. Cignoni, P.; Callieri, M.; and Corsini, M.; Dellepiane, M.; Ganovelli, F.; Ranzuglia, G. Proceedings of Sixth Eurographics Italian Chapter Conference; Fisciano, Italy, 2008; pp. 129–136.
45. Tama, F.; Gadea, F. X.; Marques, O.; Sanejouand, Y. H. Proteins Struct Funct Genetics 2000, 41, 1.
46. Li, G. H.; Cui, Q. Biophys J 2002, 83, 2457.
47. Tama, F.; Miyashita, O.; Brooks, C. L.; III. J Mol Biol 2004, 337, 985.
48. Mouawad, L.; Perahia, D. Biopolymers 1993, 33, 599.
49. Maschho, K.; Sorensen, D. A portable implementation of ARPACK for distributed memory parallel architectures. Preliminary Proceedings, Copper Mountain Conference on Iterative Methods; Copper Mountain, CO, 1996.