A Comparison of Two Equivalent Real Formulations for Complex-Valued Linear Systems Part 2: Results

Abnita Munankarmy and Michael A. Heroux
Department of Computer Science
College of Saint Benedict
37 South College Avenue
St. Joseph, Minnesota 56374 USA

Received May 14, 2002 *Accepted October 22, 2002*

ABSTRACT

Many iterative linear solver packages focus on real-valued systems and do not deal well with complex-valued systems, even though preconditioned iterative methods typically apply to both real and complex-valued linear systems. Instead, commonly available packages such as *PETSc* and *Aztec* tend to focus on the real-valued systems, while complex-valued systems are seen as a late addition. At the same time, by changing the complex problem into an equivalent real formulation (ERF), a real valued solver can be used. In this paper we consider two ERF's that can be used to solve complex-valued linear systems. We investigate the spectral properties of each and show how each can be preconditioned to move eigenvalues in a cloud around the point (1,0) in the complex plane. Finally, we consider an interleaved formulation, combining each of the previously mentioned approaches, and show that the interleaved form achieves a better outcome than either separate ERF.

[*This article is the second part of a sequence of reports. See the December 2002 issue for Part 1—Editor.*]

I. OVERVIEW OF PROBLEMS AND SOLUTION METHODS

We started our computation with small dimension matrices and then proceeded to larger dimensioned matrices [1]. For the larger matrices, our computational problems come from three application areas, namely, molecular dynamics, fluid dynamics, and electromagnetic models. The first two problems listed in Table 1 have simple tridiagonal matrices. The next three are diagonal matrices that we generated in a special way: start a diagonal matrix with all real values and then change one value at a time to a complex value until all diagonal values are complex. Patterns were then sought for the rate of convergence.

Figures 1 and 2 show the differences in iteration count of the different formulations [1]. Figures 1 and 2 involve matrices with consecutive diagonal values such as (1, 2, 3,…, 99, 100). Here, the K1 formulation started with smaller iterations and then increased to larger iterations, which means that when all the diagonal values were complex, the K1 solution was not as attractive as when they were real. The K4 formulation behaves the other way. It starts with larger iterations and then decreases as more diagonal values become complex. Hence, it gave a better solution for complex values. Figure 3 shows the graph of different iteration counts of the different formulations where the diagonal values were non-consecutive. Similar reactions occurred in this case, however, the K14 formulation gave a constant solution that was stable. Figure 3 shows the stability of matrix K_{14}.

The sixth and seventh problems listed in Table 1 come from data sets whose detailed applications we are unfamiliar with. The last three problems come from electromagnetic models. The main use of these problems to see how diagonal values affected the K14 formulation.

Problem	Dimension	Number of Non-zeros	Description
B	10	28	Simple tridiagonal matrix
B3	10	26	Simple tridiagonal matrix
M100	100	100	100 by 100 diagonal matrix with consecutive diagonal values from 1 to 100
M25	25	25	25 by 25 diagonal matrix with consecutive diagonal values from 1 to 25
Md25	25	25	25 by 25 diagonal matrix with values (1, 2, 3,…, 12, 13, 12, 11,…, 2, 1)
M3D2	1024	12480	Computational Chemistry Model I, Sherry Li, LBL/NERSC
M4D2	10000	127400	Computational Chemistry Model II, Sherry Li, LBL/NERSC
Vm214img0	1554	39378	Electromagnetic Models
Vm214img1	1554	39378	Electromagnetic Models
Vm214img45d	1554	39378	Electromagnetic Models

Table 1. Test Problem Descriptions.

MATLAB results were obtained using Version 5.3.1 [2]. In particular, we used the built-in functions `luinc`, which we have described in Section VII of our previous paper [1], `gmres` and `gmres1`. `luinc` computes an incomplete LU factorization of a given sparse matrix. It performs the incomplete LU factorization of the given matrix with drop tolerance that is a non-negative scalar. The drop tolerances we used were 0, 1.0 $e^{-}2$, and 1.0 $e^{-}3$. `gmres` or `gmres1` solves a linear system using the generalized minimum residual method (GMRES) with preconditioning provided by the ILU preconditioner computed by `lunic` [3]. We also used a diagonal preconditioner for problems M3D2, M4D2, vm214img45 and vm214img45d, where the ILU preconditioner did not help determine which formulation would be best [4].

II. RESULTS

For the first set of results in Table 2a, that is, problems B and B3, we found that smaller dimensioned matrices, when real values are larger than imaginary values, then the matrices K_1, K_4, and K_{14} all converged to a solution with the same iterations. We thus could not prefer one formulation to another. Since these were only 10 by 10 matrices, we did not bother to see how the preconditioner would affect the results.

The matrices of problems M100, M24, and Md25 were all started with all real diagonal values, with a subsequent change to imaginary one at a time. Tests were performed for each matrix when all the diagonal values were complex. Figures 1, 2, and 3 show the changes in iteration with different diagonal values. For M100, with all the diagonal values real, the K1 formulation was better than the K4 formulation; however, the K4 was better than the K1 when all the diagonal values were complex. Similar outcomes were found in the cases of the other two problems, M25 and Md25.

In general, our results show that the K14 formulation gives a better solution than K1 or K4 when complex linear problems are solved using ERF's. In fact, in these cases the K1 and K4 formulations give the same results. This point is amplified when we notice that matrices K_1 and K_4 are symmetric. The result of using matrix K_{14} were consistent for any diagonal values. It did not matter how many real or complex values were in the diagonal of the matrix.

For problem M3D2, we computed the spectrum of the original and the preconditioned matrices using the MATLAB `luinc` function. Figure 4 shows the distribution of the eigenvalues of the original complex matrix and Figure 5 shows the

Problem	***Tolerance***	***C Iterations***	***K_1 Iterations***	***K_4 Iterations***	***K_{14} Iterations***
B	$1 \times e^{-6}$	10	20	20	20
B3	$1 \times e^{-6}$	10	20	20	20
M100	$1 \times e^{-6}$	47	200	47	NA
M25	$1 \times e^{-6}$	22	50	22	22
Md25	$1 \times e^{-6}$	13	26	13	13
vm214img0	$1 \times e^{-6}$	268	590	NA	590
vm214img1	$1 \times e^{-6}$	134	809	NA	809
vm214img45	$1 \times e^{-6}$	37	243	70	70

Table 1. MATLAB test results using GMRES (∞) without preconditioning.

Problem	***Tolerance***	***C Iterations***	***K_1 Iterations***	***K_4 Iterations***	***K_{14} Iterations***
M3d2	$1 \times e^{-12}$	495	1434	NA	1180
vm214img45	$1 \times e^{-12}$	65	465	128	128
vm214img45d	$1 \times e^{-12}$	65	477	427	128

Table 2. MATLAB test results using GMRES (∞) with Diagonal Preconditioning.

Problem	***Tolerance***	***C Iterations***	***K_1 Iterations***	***K_4 Iterations***	***K_{14} Iterations***
M4d2	$1 \times e^{-3}$	44	74	74	74

Table 3. MATLAB test results using GMRES (∞) with `luinc` (droptot) Preconditioning.

eigenvalues of the K matrix. As expected, the eigenvalues of the K formulation matrix are the eigenvalues of the complex matrix plus their reflection about the real axis. For this problem, the ILU preconditioner did not help us distinguish between the formulations, so the diagonal preconditioner was used. The number of iterations by formulation is shown in Table 3. In general, the K14 formulation gave the better answer. However, the comparison was not complete since the K4 formulation was not available. (The matrix K_4 was singular, with diagonal elements of zero, and thus not appropriate for use in preconditioning.) For problem M4d2 in Table 3, we could not use the diagonal preconditioner because of the large matrix size, and so the luinc preconditioner instead. Table 3 shows that all formulations have the same result, something that we believe cannot be true. We think that this similarity arises due to insufficient memory.

The last three problems in Table 1 were interesting to observe. For vm214img0, real values were larger than imaginary; hence the matrix K_{14} was identical to K_1. Obviously the results for these two methods should be the same, and that is what we found (Table 1). The matrix K_4 could not be test because of its zeros along the diagonal. The preconditioner did not work in this problem because of the non-structured array of a unit lower triangular matrix. We should note that the preconditioner fails if the diagonal has values of the unit lower or upper triangle have any zeros. For vm214img1, a similar result was observed. However, for vm214img45, the results were different. Here the diagonal values of the complex matrix had its imaginary part larger than its real part. Hence the matrix K_{14} was similar

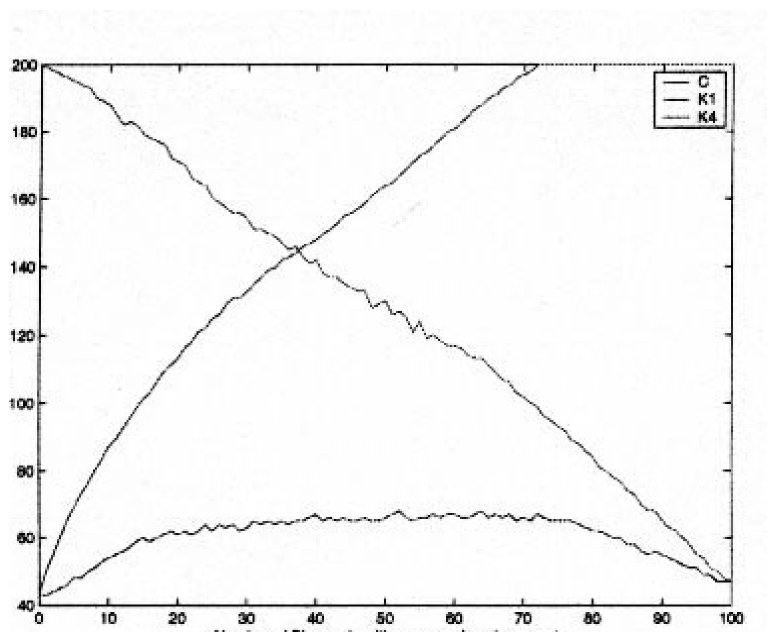

Figure 1. Graph of Iteration Count (vertical axis) versus Number of Diagonal Terms with Non-Zero imaginary parts (horizontal axis) for each of the formulations, using a 100 × 100 diagonal matrix with consecutive values for the diagonal elements, as described in the text.

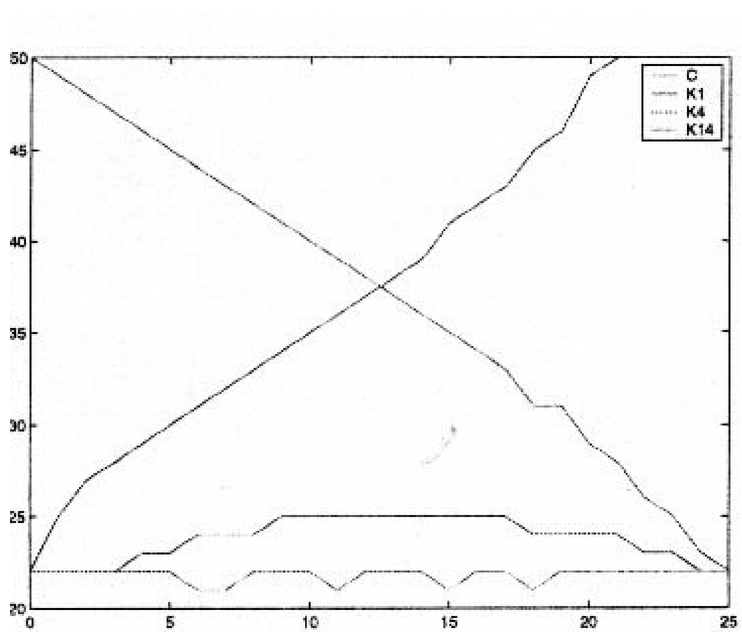

Figure 2. Graph of Iteration Count (vertical axis) versus Number of Diagonal Terms with Non-Zero imaginary parts (horizontal axis) for each of the formulations, using a 25 × 25 diagonal matrix with consecutive values for the diagonal elements, as described in the text.

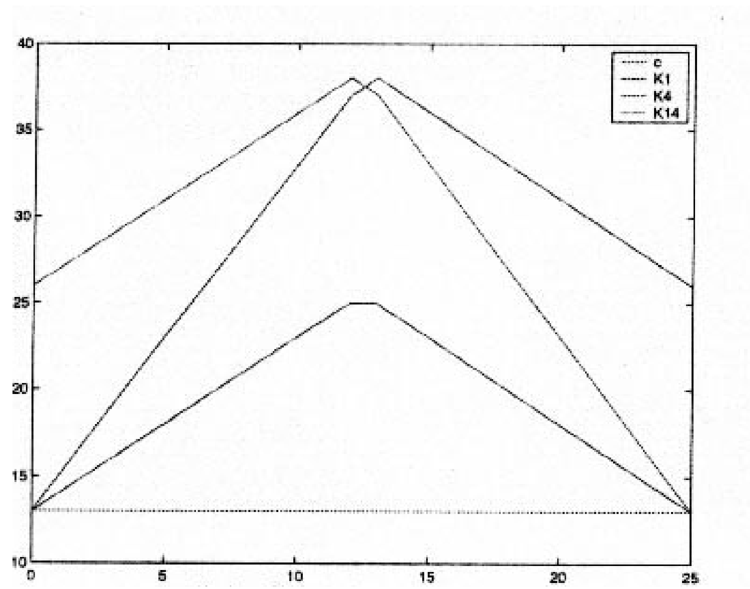

Figure 3. Graph of Iteration Count (vertical axis) versus Number of Diagonal Terms with Non-Zero imaginary parts (horizontal axis) for each of the formulations, using a 25 × 25 diagonal matrix with non-consecutive values for the diagonal elements, as described in the text.

to matrix K_4. We could use a preconditioner get better results. We used a diagonal preconditioner, and Tables 1 and 2 show the results. With the preconditioner, K4 was found to be better than K1 (see Table 2). K14 was similar to K4 as the matrices produced from these formulations were similar. However, to see how much better the K14 formulation would be, we made a small change in the original problem vm214img45, naming this new file vm214img45d. We changed half the diagonal entries in the complex matrix C such that half of them now had real parts larger than their imaginary parts. The results of problem vm214img45d clearly show that formulation K14 is better than formulation K1 or formulation K4 used alone (see Table 2). C1 is the name we gave this new C matrix generated in the adjustment to problem vm214img45 (see below). It should be compared with matrix C of section III of Munankarmy and Heroux (2002) [1].

$$C1=\begin{bmatrix} 2+i & 2+i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1+\frac{1}{2}i & 2+4i & 3+\frac{3}{2}i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2+i & 3+6i & 4+2i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3+\frac{3}{2}i & 4+8i & 5+\frac{5}{2}i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4+2i & 5+10i & 6+3i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5+\frac{5}{2}i & 12+6i & 7+\frac{7}{2}i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6+3i & 14+7i & 8+4i & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7+\frac{7}{2}i & 16+8i & 9+\frac{9}{2}i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8+4i & 18+9i & 10+5i \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9+\frac{9}{2}i & 20+10i \end{bmatrix}$$

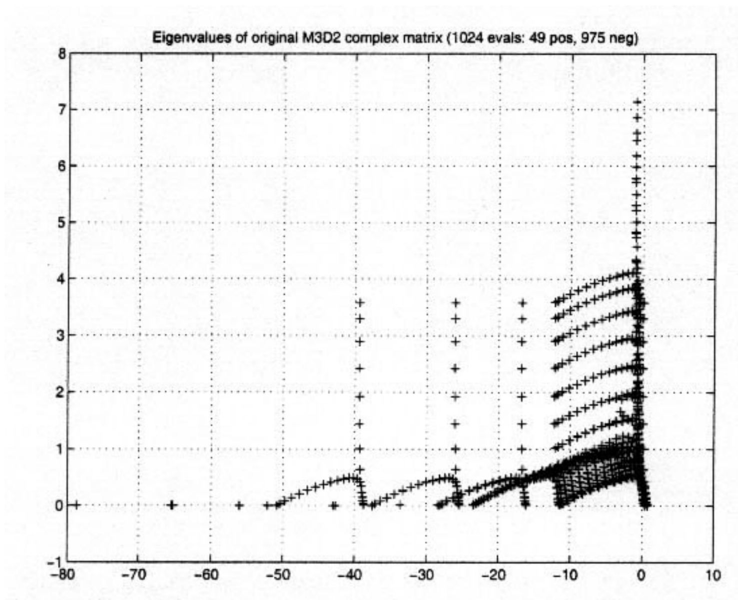

Figure 4. Eigenvalues of the original complex matrix in problem M3D2, with imaginary components graphed on the vertical axis and real parts of the eigenvalues graphed on the horizontal axis.

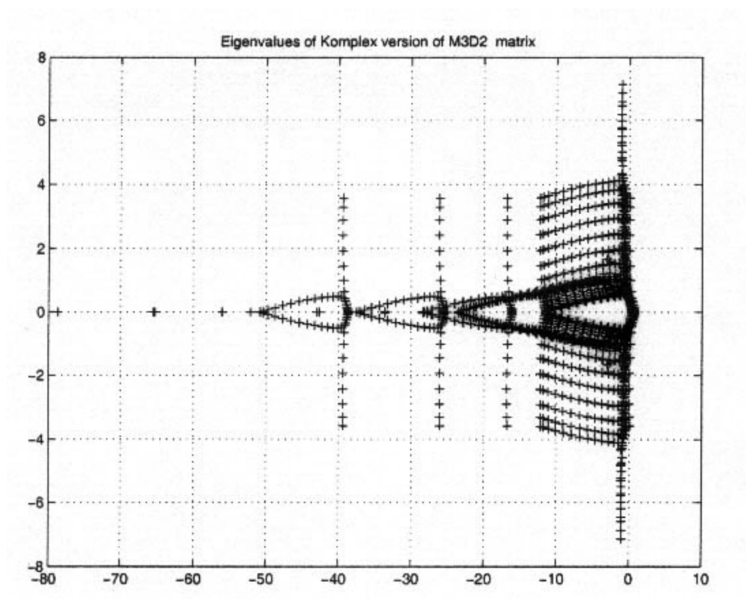

Figure 5. Eigenvalues of the K formulation matrix in problem M3D2, with imaginary components graphed on the vertical axis and real parts of the eigenvalues graphed on the horizontal axis.

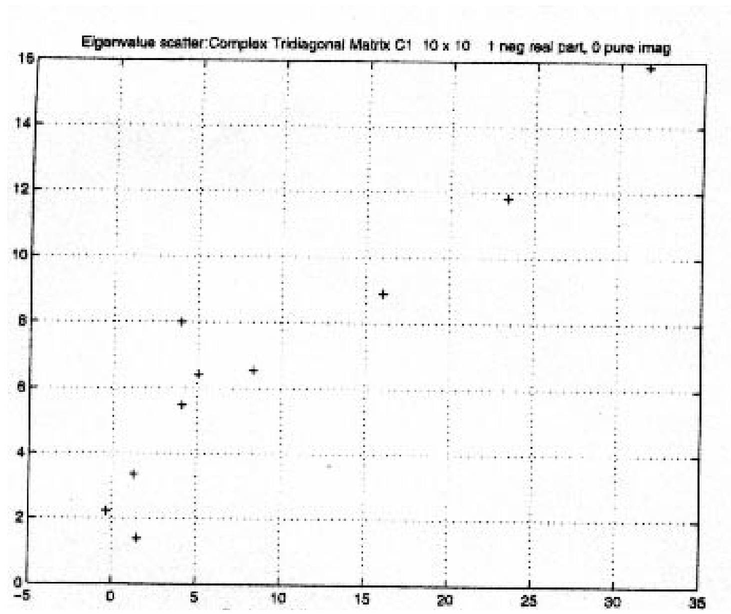

Figure 6. Eigenvalues of the new complex tridiagonal matrix C1, with imaginary components graphed on the vertical axis and real parts of the eigenvalues graphed on the horizontal axis.

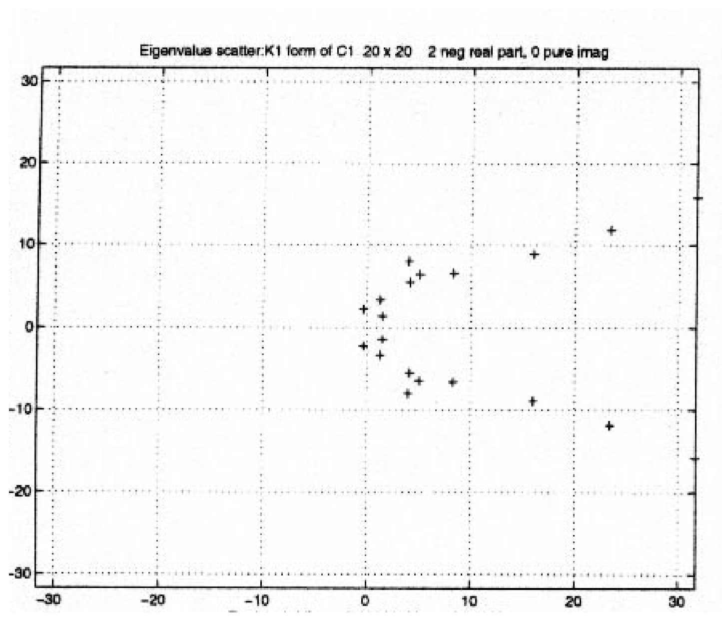

Figure 7. Eigenvalues of the K1 formulation matrix using C1, with imaginary components graphed on the vertical axis and real parts of the eigenvalues graphed on the horizontal axis.

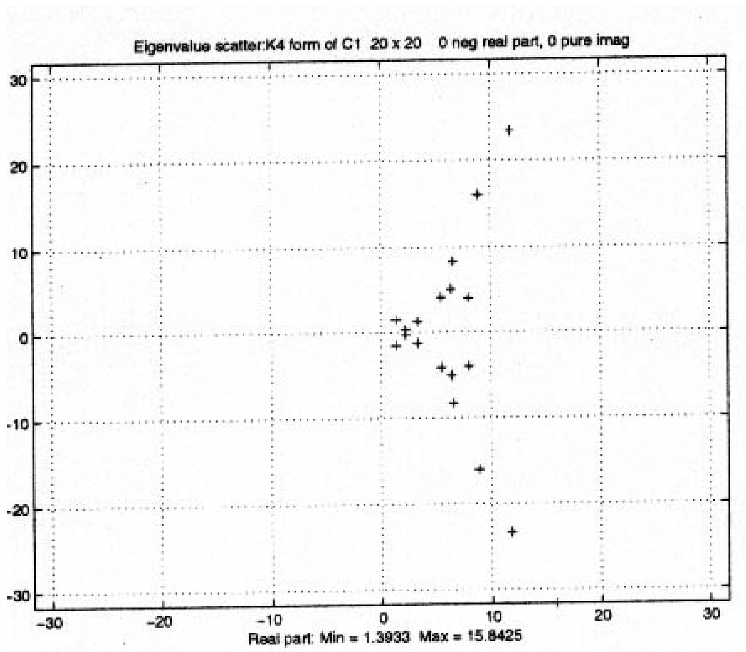

Figure 8. Eigenvalues of the K4 formulation matrix from C1, with imaginary components graphed on the vertical axis and real parts of the eigenvalues graphed on the horizontal axis.

III. CONCLUSIONS

In this report and the previous paper [1], we presented a comparison of two real equivalent formulations for complex-valued linear systems. In addition, we also presented what we termed the interleaved formulation, K14. Our results show that in cases where the imaginary components of matrix elements are larger than their respective real parts, then the K4 formulation provides a more efficient method of solution.

Challenging problems require a high-quality preconditioner for rapid convergence. Such preconditioners move eigenvalues in a cloud around point (1,0) in the complex plane. This shows that the requirement of a high-efficiency preconditioner provides the best solution and diminishes the convergence differences between a true complex iterative solver and the K1, K4, and K14 formulations. Finally, our results indicate that formulation K14 is at least as efficient as either the K1 or K4 formulations. More research is needed before a stronger statement can be made.

REFERENCES

1. Abnita Munankarmy and Michael A. Heroux, “A Comparison of Two Equivalent Real Formulations for Complex-Valued Linear Systems, Part I: Introduction and Method” *Am. J. Undergrad. Res.*, Vol. 1, No. 3 (December 2002), pp. 17-26.
2. MathWorks. See the MATLAB home-page at http:// www.mathworks.com
3. Youcef Saad and Martin H. Schultz, ‘GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM J. Sci. Statist. Comput.*, Vol. 7, No. 3 (july 1986), pp. 856-869.
4. M. Baertschy, T. N. Rescigno, W. A. Issacs, X. Li, and C. W. McCurdy, "Electron-impact ionization of atomic hydrogen", Physical Review A, Vol. 63, p. 22712 (January 18, 2001).