REPORT NO.
UCB/SESM-80/08

STRUCTURAL ENGINEERING AND
STRUCTURAL MECHANICS

LANCZOS VERSUS SUBSPACE ITERATION FOR SOLUTION OF EIGENVALUE PROBLEMS

by
BAHRAM NOUR-OMID
BERESFORD N. PARLETT
and
ROBERT L. TAYLOR

Submitted to:
International Journal of Numerical Methods in Engineering

NOVEMBER 1980

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA

LANCZOS VERSUS SUBSPACE ITERATION FOR SOLUTION OF EIGENVALUE PROBLEMS

by

Bahram Nour-Omid*
Beresford N. Parlett**
Robert L. Taylor*

Abstract: The two algorithms are tested on two structural problems and compared. The Lanczos method is too good to ignore.

* Civil Engineering Department, University of California, Berkeley, California 94720, U.S.A.

** Mathematics Department, University of California, Berkeley, California 94720, U.S.A.

1. Introduction

Subspace iteration has become the standard tool for solving the large eigen-problems which occur in the dynamic analysis of structures, typically

$$[\underset{\sim}{K} - \lambda \underset{\sim}{M}]\ \underset{\sim}{\phi} = 0 \qquad (1)$$

where $\underset{\sim}{K}$ and $\underset{\sim}{M}$ are respectively the stiffness and mass matrices and $(\lambda, \underset{\sim}{\phi})$ is an eigen-pair.

Our goal is not to do a definitive comparison of subspace iteration with the Lanczos algorithm but simply to point out that, when used properly, Lanczos is too good to ignore. Our comparison (Figs. 2 and 4) stopped when subspace iteration exhausted our resources. Out of core solutions are discussed in Section 4.

It is all too easy to load the dice, even inadvertantly, against one side in a comparison of numerical methods. We tried to be aware of this throughout the study.

2. Lanczos

The Lanczos algorithm is less familiar to structural engineers than is subspace iteration. A full description can be found in ref. (1).

Here we will just say that the simple Lanczos algorithm behaves as though it retained - and used to a great advantage - every vector computed in a run of the power method (Stodola's iteration). In fact this information is coded in a tridiagonal matrix and it is some of the eigen-values of this tridiagonal matrix which converge to the eigen-values of the big matrix. Convergence of the outer eigen-values occurs surprisingly fast. Storage needs for Lanczos are considerably less than for subspace iteration.

There was a defect in the Lanczos algorithm which may have prevented its acceptance: when should it be stopped? In contrast to subspace iteration there is no need to compute the smallest p eigen-values of the tridiagonal $\underset{\sim}{T}$ at each step. Even worse, only some of the eigen-values of $\underset{\sim}{T}$ approximate those of $(\underset{\sim}{K}, \underset{\sim}{M})$. These blemishes have been overcome (see chapter 13 in ref 1) and now the Lanczos iteration will be halted at the earliest possible step. The user does not have to get involved beyond designating the desired accuracy in the eigen-vectors. Of course, the eigen-values will be far more accurate.

Block versions of Lanczos are more complicated and more powerful, just as subspace iteration is more powerful than simple inverse iteration. Naturally block Lanczos produces a block tridiagonal matrix. However, the fact that Lanczos can be used with blocks of any convenient size must not distract attention from the power of Lanczos with blocks of size 1, 2, or 3.

The program we obtained was aimed at big problems ($n > 1000$) and it allows the user to choose the block size. The smallest effective size is 2, and this is what we used on our small problems. The results would have been still more favorable to Lanczos had we used a simple version.

We turn now to an important misconception concerning the Lanczos algorithm. It is usually presented as a way to compute eigen-values and eigen-vectors of a symmetric matrix $\underset{\sim}{A}$. Although the algorithm can be formulated to work on a pair $(\underset{\sim}{K}, \underset{\sim}{M})$, some form of inversion or factoring of a matrix, either explicitly or implicitly, is required. If $\underset{\sim}{M}$ is positive definite and of narrow bandwidth then the usual recommendation is to factor $\underset{\sim}{M}$ once and for all as $\underset{\sim}{M} = \underset{\sim}{L}_1 \underset{\sim}{L}_1^T$ (this is the Choleski factorization, not the square root) and then take $\underset{\sim}{L}_1^{-1} \underset{\sim}{K} (\underset{\sim}{L}_1^{-1})^T$ as $\underset{\sim}{A}$. On the other hand it is always possible to factor $\underset{\sim}{K}$ into $\underset{\sim}{L}_2 \underset{\sim}{L}_2^T$ and then find the largest few eigen-vlaues of $\underset{\sim}{L}_2^{-1} \underset{\sim}{M} (\underset{\sim}{L}_2^{-1})^T$ [of course the products given above are not computed explicitly because that would destroy sparsity].

There has been no mention of shifts and it is sometimes asserted that Lanczos does not or cannot take advantage of good shifts. The truth is a bit more subtle. Lanczos is so powerful that, indeed, the eigen-value problem can be solved efficiently with either $\underset{\sim}{L}_1^{-1} \underset{\sim}{K} (\underset{\sim}{L}_1^{-1})^T$ or $\underset{\sim}{L}_2^{-1} \underset{\sim}{M} (\underset{\sim}{L}_2^{-1})^T$. Nevertheless if it is possible to factor $\underset{\sim}{K} - \sigma\underset{\sim}{M}$ for a well chosen value of σ then it pays handsomely to run Lanczos with

$$\underset{\sim}{A} = \underset{\sim}{M}^{\frac{1}{2}} (\underset{\sim}{L}_\sigma^{-1})^T \underset{\sim}{D}^{-1} \underset{\sim}{L}_\sigma^{-1} \underset{\sim}{M}^{\frac{1}{2}} \tag{2}$$

where

$$\underset{\sim}{K} - \sigma\underset{\sim}{M} = \underset{\sim}{L}_\sigma \underset{\sim}{D} \underset{\sim}{L}_\sigma^T \tag{3}$$

Note that M need not be positive definite.

This important way of shifting was introduced by Ericsson and Ruhe in ref. (3) as the spectral transformation. The eigen-values α_i of $\underset{\sim}{A}$ are related to the eigen-values λ_i of ($\underset{\sim}{K}$, $\underset{\sim}{M}$) by

$$\alpha_i = 1/(\lambda_i - \sigma) \tag{4}$$

The message is that although shifts can be used, fewer of them are needed than with subspace iteration.

There were defects in the Lanczos algorithm which may have delayed its acceptance. The complete loss of orthogonality among the Lanczos vectors is not a disaster. It serves only to delay convergence somewhat beyond the theoretical optimum, not to prevent it. This was the contribution of Paige's thesis in [1971]. The technique of selective orthogonalization keeps the Lanczos vectors robustly linearly independent and keeps the number of iterations close to the minimum. See ref. (6) for more details. More troublesome is, or was, the choice of the right moment to stop. It is overkill to compute all the eigen-values of the tridiagonal matrix at each step. Only a few are needed. See ref. (1) for more details.

3. Subspace Iteration

The subspace iteration is widely used by structural engineers for extracting eigen-pairs of dynamic systems. A detailed description can be found in ref. (4) and (5). This method can be implemented in a number of different ways. The version used to solve the generalized eigen-problem is the one developed by Bathe & Wilson and implemented within FEAP. Other versions of the subspace iteration algorithm are in use but the costs for these variations are very close to the one used.

4. Storage Requirements

For large problems it is essential to use secondary storage devices (such as discs or magnetic tapes) and the variety of computing systems makes it difficult to generalize about their use.

As far as we can see, the use of secondary storage favors Lanczos even more strongly than do our in-core results. There are a significant number of intermediate size problems in which the factors of $\underset{\sim}{K}$ and $\underset{\sim}{M}$ can be held in core but not all the subspace vectors. All Lanczos needs is to write out an old Lanczos vector at each step and then, from time to time, to read in all the Lanczos vectors successively. In addition to the factors of $\underset{\sim}{K}$ and $\underset{\sim}{M}$, Lanczos must hold in core only the tridiagonal matrix, two working vectors, and space for perhaps half of the wanted eigen-vectors. In contrast, subspace iteration would need to access the basis vectors at every step and to hold in core the small eigen-problem which must be solved at every step.

For very large problems the Choleski factor $\underset{\sim}{L}$ of $\underset{\sim}{K} - \sigma\underset{\sim}{M}$ will have to reside out of core and the solution of triangular systems $\underset{\sim\sim}{L}\underset{\sim}{v} = \underset{\sim}{w}$ will have to be done in pieces. It is here that the fact that Lanczos requires an almost minimal number of matrix-vector operations is most attractive. For each problem there is a special number, r say, such that the cost of computing $\underset{\sim\sim}{A}\underset{\sim}{v}$ for r vectors $\underset{\sim}{v}$ is the same as the cost for just one vector $\underset{\sim}{v}$. This facilitates the multiplication of the basis vectors in subspace iteration but, for the same reason, it favors Lanczos with blocks of size r.

In extreme cases the Lanczos algorithm can function even when only 7 or 8 vectors can be held in core at one time: 2 for the Lanczos step,

2 for the tridiagonal, 1 for selective orthogonalization, and 2 or 3 for parts of $\underset{\sim}{L}$.

5. Computer Procedures

In our study two different programs were used (almost as "black boxes"). The Lanczos algorithm was the Block version developed at Oak Ridge National Laboratory by David Scott (ref. 2). The program is designed to solve the standard eigen-problem, and was originally developed on an IBM machine. ANSI FORTRAN was used to assure portability. The software was transferred to the CDC 6400 at U.C. Berkeley. The Oak Ridge program is set to find a given number of eigen-values at either end of the spectrum (but not both). All the communication between the program and the eigen-problem is done through subroutine OP (supplied by the user), so the modifications required to extend the program to the generalized eigen-problem using $\underset{\sim}{M}^{\frac{1}{2}}(\underset{\sim}{K} - \sigma\underset{\sim}{M})^{-1}\underset{\sim}{M}^{\frac{1}{2}}$ was done through this subroutine.

The subspace iteration algorithm was the one utilized within the finite element program FEAP which was developed at U.C. Berkeley by R. L. Taylor (see chapter 24 of ref. 7). FEAP was also used to generate the stiffness and mass matrices of our test problems. We followed the standard practice in choosing as p, the subspace dimension, min (2m, m+8) where m is the number of wanted eigen-values.

The execution times were determined from calls to the clock of the computer.

6. Effect of Shifting

To determine the effect of a shifting strategy on each algorithm, a series of tests was performed in which various shifts were applied to the test problems and the C.P.U. times to extract the first three eigen-pairs were obtained. The results were plotted in figs. 5 and 6. A good shifting strategy will reduce the cost of each algorithm. However, figs. 5 and 6 show that the saving is somewhat more in Lanczos than in subspace iteration. Therefore the use of a sophisticated shifting strategy in Lanczos, as in Ericsson and Ruhe (ref. 3), would only enhance the cost ratio between the two methods.

7. Test Problems

The problems used to test and compare the above algorithms were typical problems occuring in structural analysis. They were set up in such a way as to reflect the problems encountered in very large systems. The details of the problems are layed out in figs. 1 and 3.

8. Discussion

It is evident from the results (figs. 2 and 4) that the Lanczos algorithm is of an order of magnitude faster and therefore less costly than Subspace iteration. It is never slower.

References

1. B.N. Parlett, "The Symmetric Eigenvalue Problem", Prentice-Hall, Englewood Cliffs, N.J. 1980.

2. D.S. Scott, "Block Lanczos Software for Symmetric Eigenvalue Problems", Report ORNL/CSD-48, UC-32.

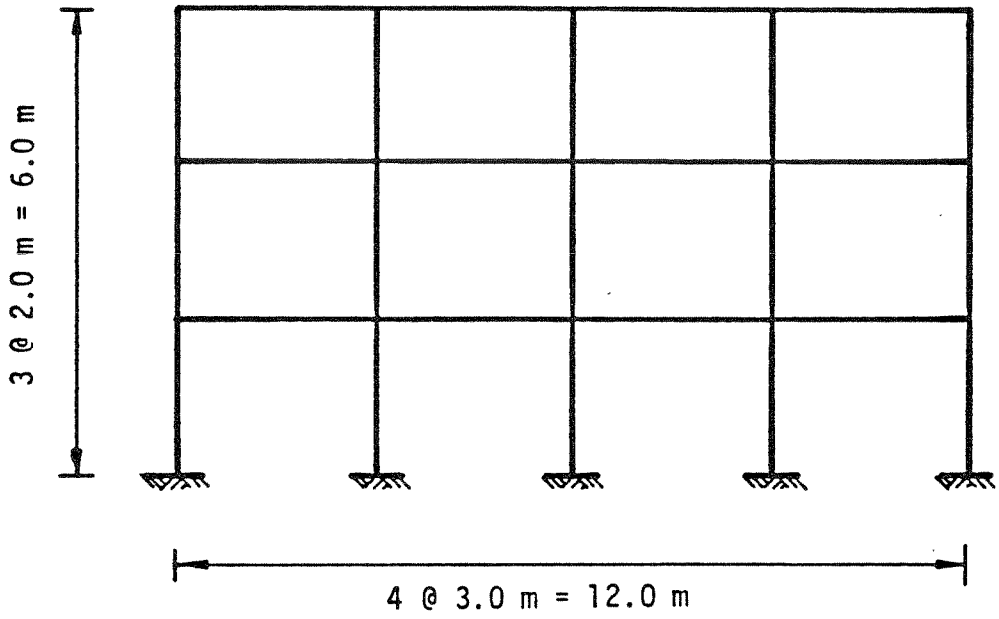
3. T. Ericsson and A. Ruhe, "The Spectral Transformation Lanczos Method for the Numerical Solution of Large Sparse Generalized Symmetric Eigenvalue Problems", Report UMINF-76.79 ISSN 0348-0542.

4. K.S. Bathe and E.L. Wilson, "Solution Methods for Eigenvalue Problems in Structural Mechanics", International Journal for Numerical Methods in Engineering, Vol. 6, pp. 213-226 (1972).

5. K.J. Bathe and E.L. Wilson, "Numerical Methods in Finite Element Analysis", Prentice-Hall, Englewood Cliffs, N.J. 1976.

6. B.N. Parlett and D.S. Scott, "The Lanczos Algorithm with Selective Orthogonalization", Mathematics of Computations, Vol. 33, number 145, Jan. 1979, pp. 217-238.

7. O.C. Zienkiewicz, "The Finite Element Method", third edition, McGraw-Hill, 1977.



For all beams and columns:

Young's Modulus = 1.0 KN/m^2

Mass Density = 1.0 Kg/m^3

Area = 1.0 m^2

Moment of Inertia = 1.0 m^4

No. of Beam Elements = 27

No. of Nodes = 20

Total No. of D.O.F. = 45

Figure 1a. Building Frame of Example 1.

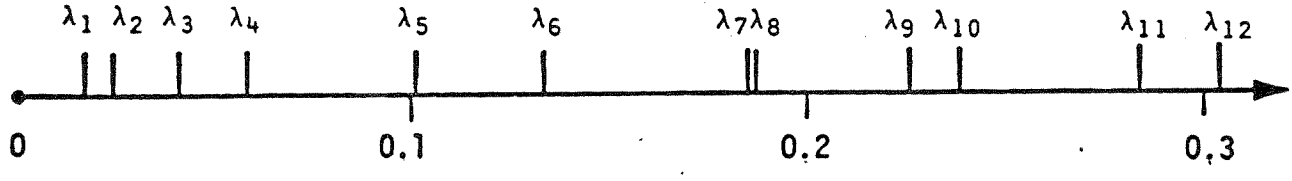


Figure 1b. Eigenvalues of the Above System

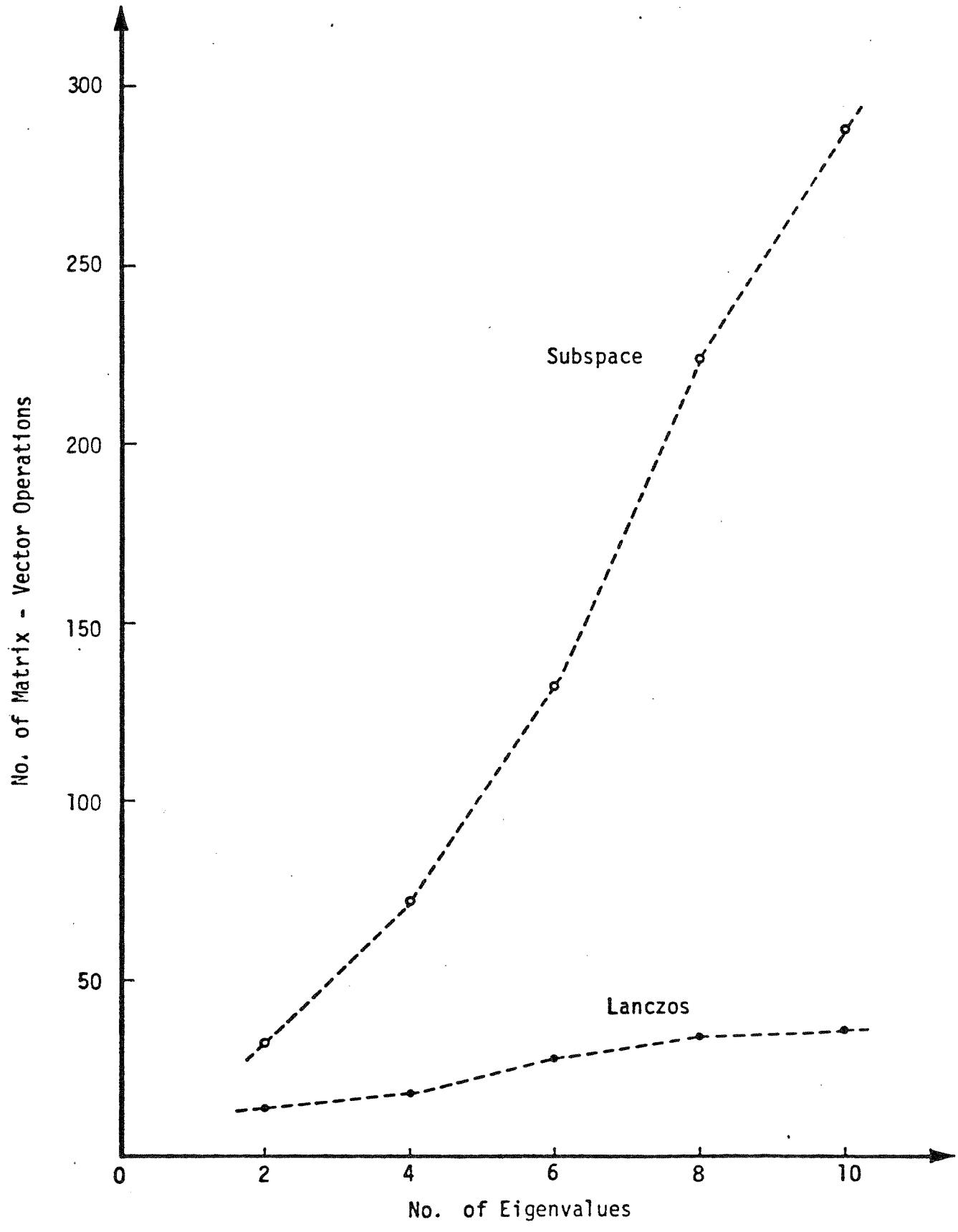


Figure 2.a. Comparison of Matrix - Vector Operations for Obtaining Increasing Numbers of Eigenpairs for Building Frame of Example 1.

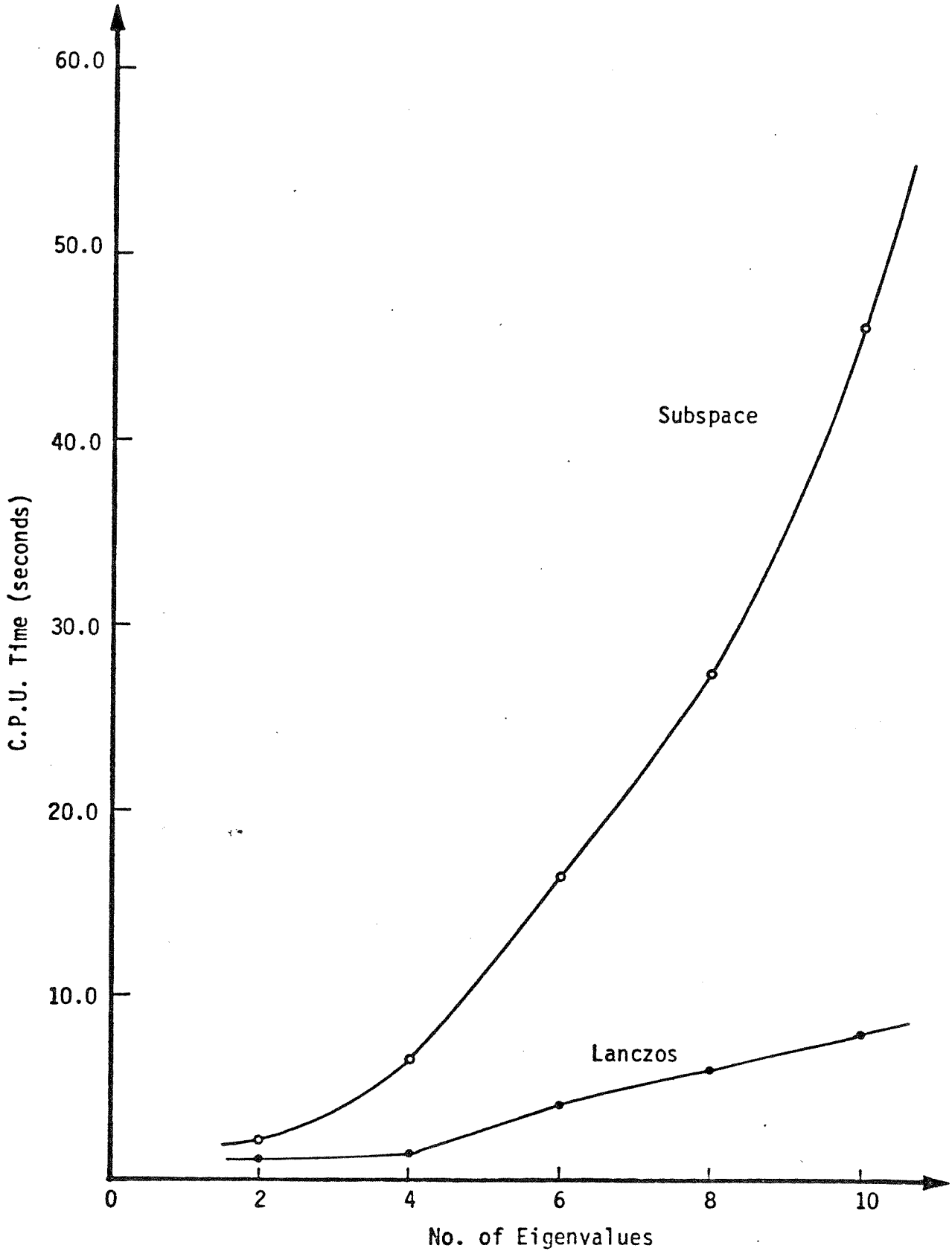
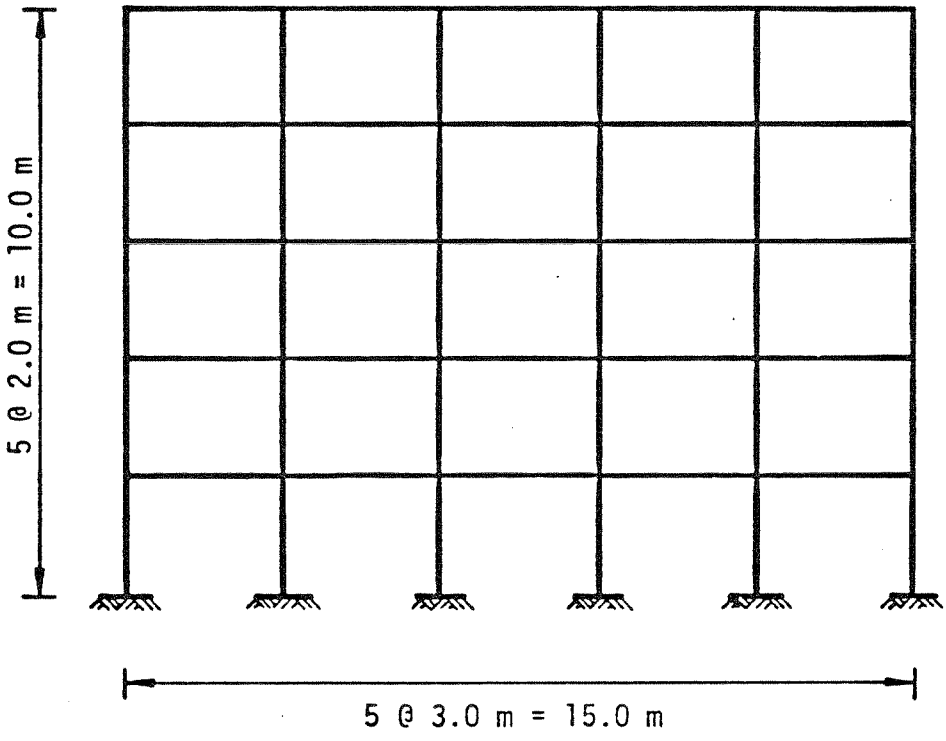


Figure 2b. Comparison of Solution Times for Obtaining Increasing Numbers of Eigen pairs for Building Frame of Example 1.



For all beams and columns:

Young's Modulus = 1.0 KN/m^2

Mass Density = 1.0 Kg/m^3

Area = 1.0 m^2

Moment of Inertia = 1.0 m^4

No. of Beam Elements = 55

No. of Nodes = 36

Total No. of D.O.F. = 90

Figure 3a. Building Frame of Example 2.

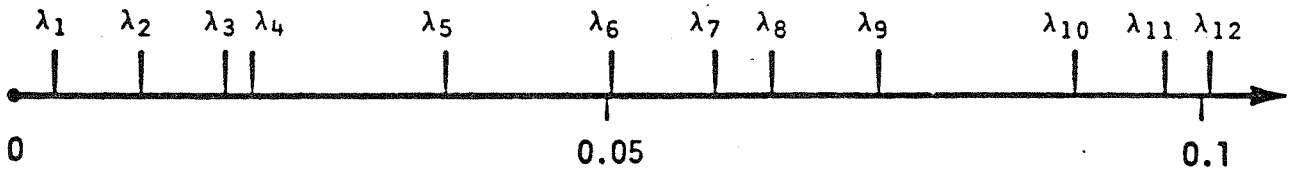


Figure 3b. Eigenvalues of the Above System

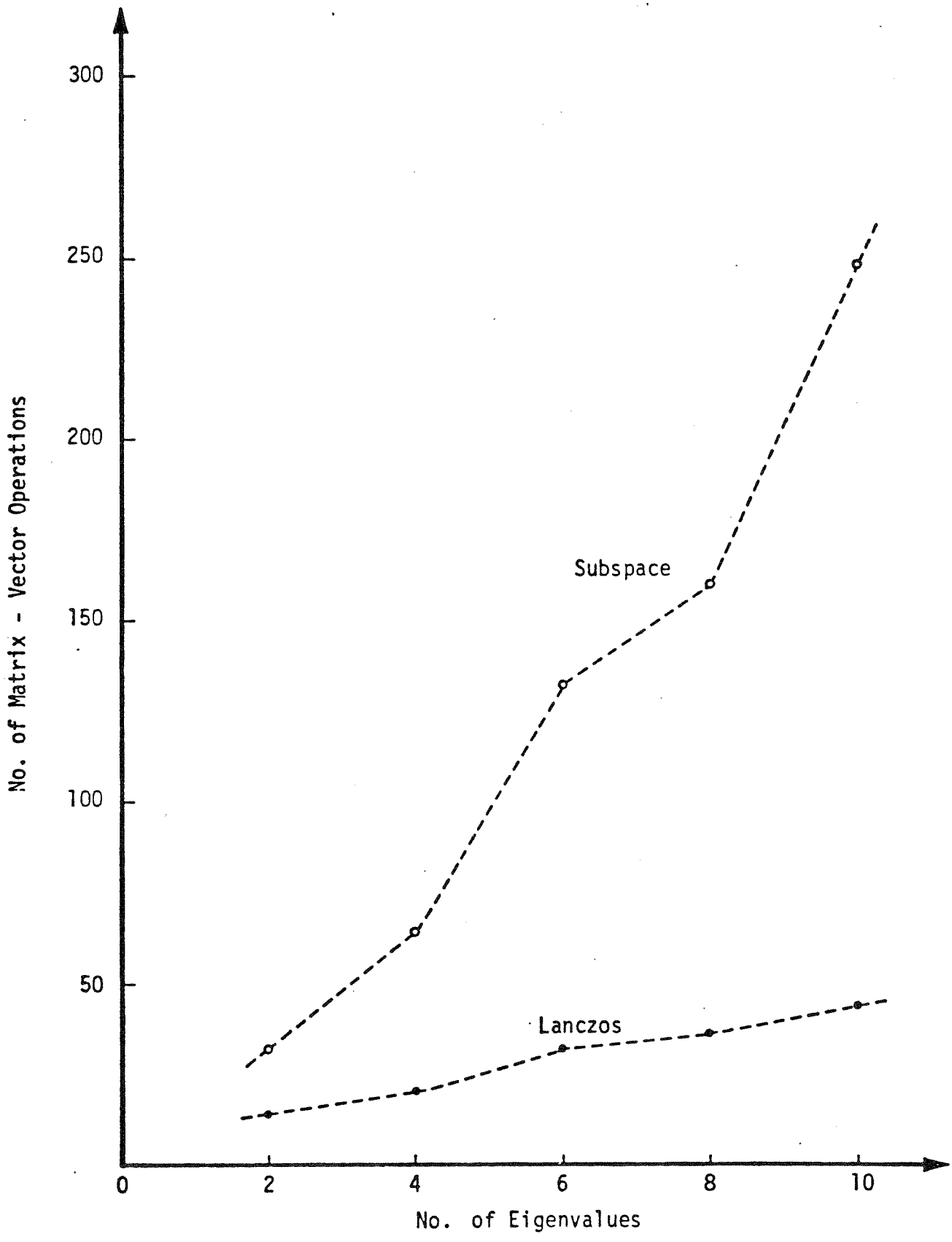


Figure 4.a. Comparison of the Matrix - Vector Operations for Obtaining Increasing Numbers of Eigenpairs for Building Frame of Example 2.

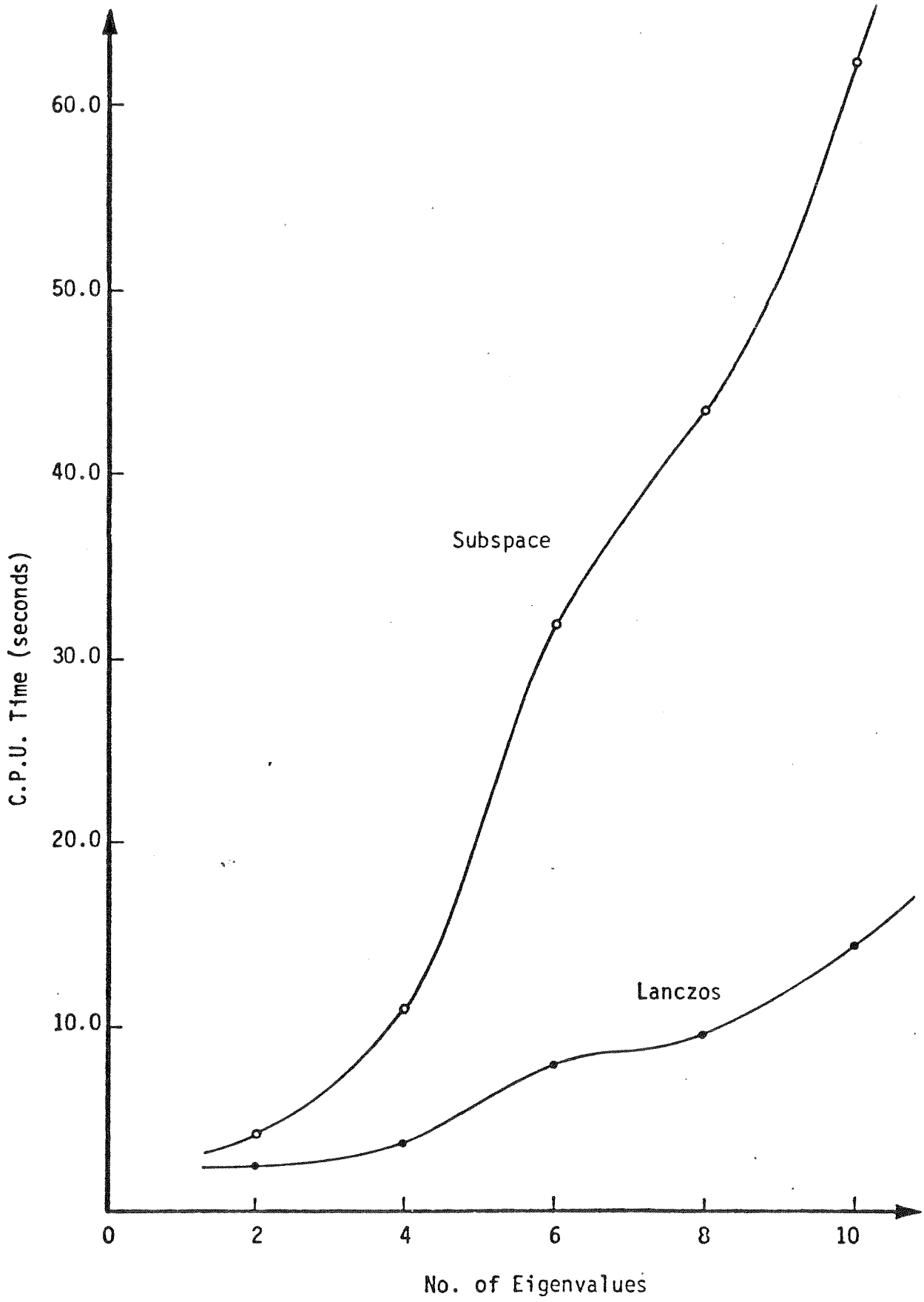


Figure 4b. Comparison of Solution Times for Obtaining Increasing Number of Eigen pairs for Building Frame of Example 2.

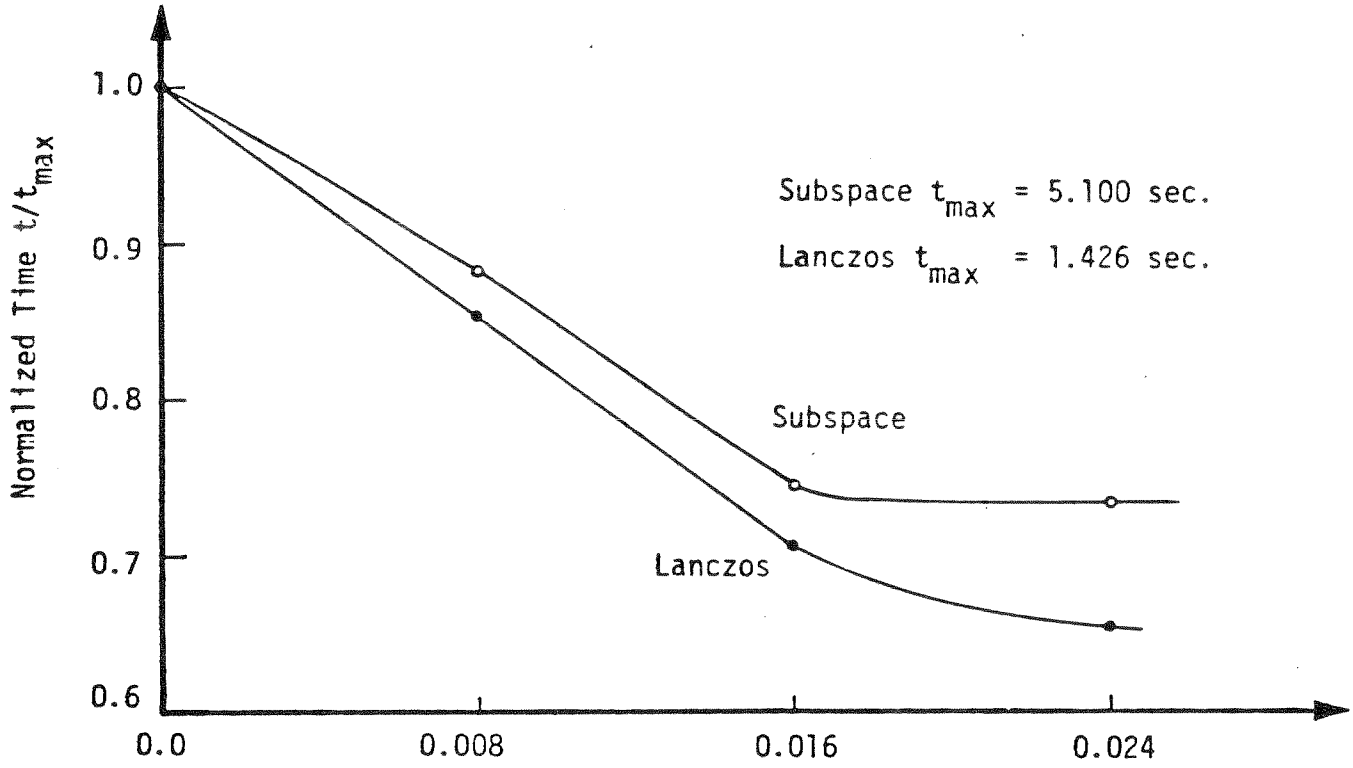


Figure 5. Results from Example 1.

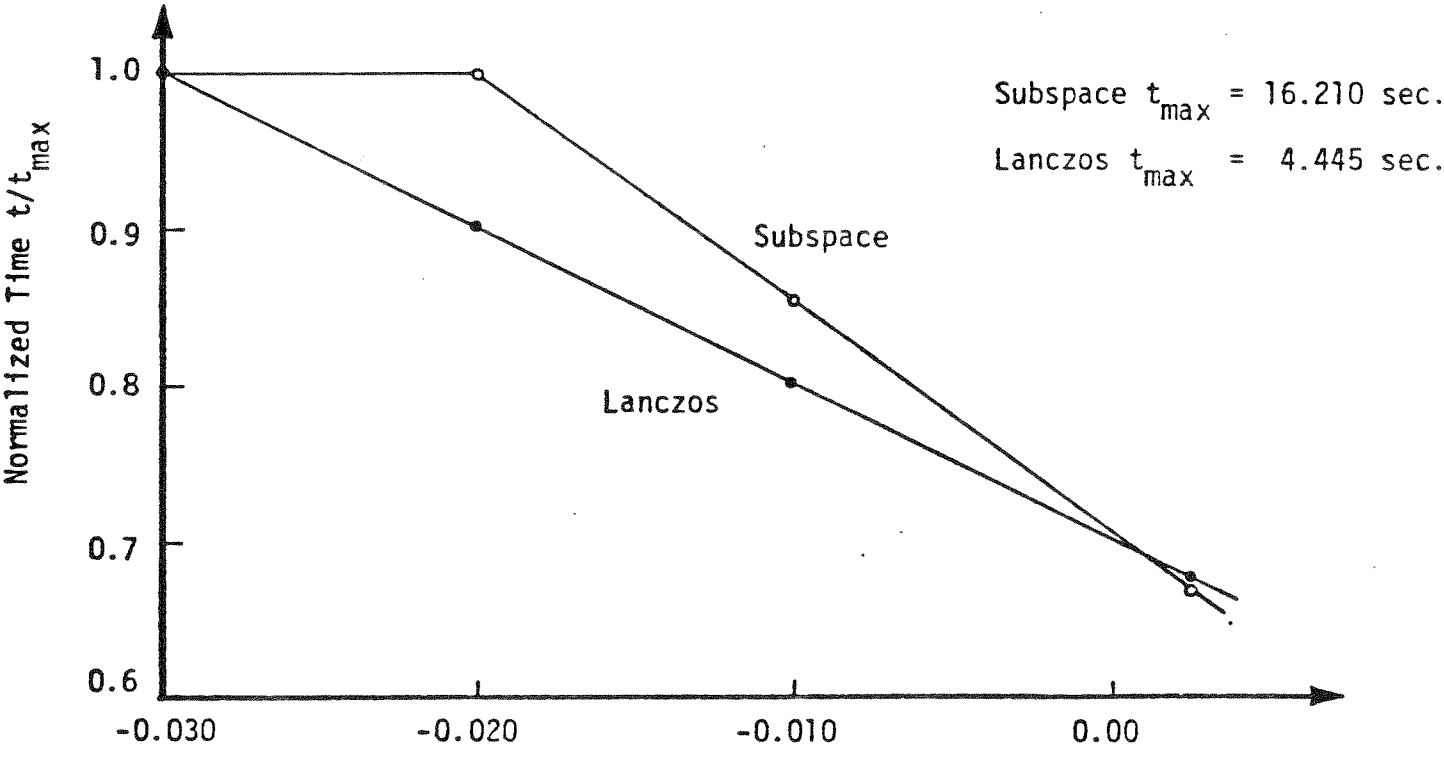


Figure 6. Results from Example 2.

Relative Improvement of Solution Times for Obtaining