

International Scholarly Research Network
ISRN Applied Mathematics
Volume 2011, Article ID 236727, 6 pages
doi:10.5402/2011/236727

Research Article

A Solution Approach for Lower Hessenberg Linear Systems

Anthony A. Ruffa

Chief Technology Office, Naval Undersea Warfare Center, 1176 Howell Street, Newport, RI 02841, USA

Correspondence should be addressed to Anthony A. Ruffa, anthony.ruffa@navy.mil

Received 28 October 2011; Accepted 17 November 2011

Academic Editors: P. J. García Nieto and F. Lebon

Copyright © 2011 Anthony A. Ruffa. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An approach is developed to obtain solutions to lower Hessenberg linear systems with general entries. The approach involves developing solution vectors for an extended lower Hessenberg linear system (having an extra column and an extra introduced unknown) for each nonzero term on the right hand side. The overall solution is then found through superposition and determination of the extra introduced unknown. The approach supports parallel solution algorithms without communication between processors, since each solution vector is computed independently of the others. The number of parallel processors needed will be equal to the number of nonzero right hand side terms.

1. Introduction

A number of researchers have studied the inverses of lower Hessenberg matrices [1–4], that is, inverses of square matrices of the form $A = (a_{ij})$, $i, j = 1, \dots, n$ such that $a_{ij} = 0$ when $j > i + 1$. Most recently, a recursive algorithm has been developed for inverting Hessenberg matrices [1]. This paper proposes an alternate solution approach. It is shown here that lower Hessenberg linear systems in particular lend themselves to a solution via an extended system (that adds a column to A , as well as an additional unknown). The basic strategy involves generating solution vectors for m such extended systems (where m is the number of nonzero right hand side terms). In each such extended system, all of the right hand side terms are set to zero except for one entry. The first term of each solution vector is chosen arbitrarily, and then each subsequent term is found directly through a forward-substitution-like process similar to that used in LU decomposition, with a number of operations on the order of that required for forward substitution, that is, an order of magnitude smaller than that required for performing the LU decomposition itself. The overall solution is found through superposition and solution of the extra introduced variable (which is common to all of the m extended

systems). The process is highly parallelizable, since each solution vector can be computed independently of the others.

This approach will first be illustrated for a three-dimensional system (i.e., $n = 3$) with general entries in Section 2. Section 3 provides a proof of the validity of the approach for systems having an arbitrary dimension n .

2. Three-Dimensional Systems

This section details the approach to solve a lower Hessenberg linear system with $n = 3$ and general entries. Consider the following system:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix}. \quad (2.1)$$

This can be solved via an extended system developed by adding an additional column to the coefficient matrix, and an additional unknown x_4 :

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 + x_4 \end{Bmatrix}, \quad (2.2)$$

where $a_{i,i+1} \neq 0$ for all i . The coefficient $a_{3,4}$ is set to 1 to simplify the solution process. Note that the third equation in (2.2) is

$$a_{3,1}x_1 + a_{3,2}x_2 + a_{3,3}x_3 + x_4 = b_3 + x_4, \quad (2.3)$$

or

$$a_{3,1}x_1 + a_{3,2}x_2 + a_{3,3}x_3 = b_3. \quad (2.4)$$

The strategy for solving (2.2) involves first solving three such systems, each with a different nonzero right hand side term to obtain the solution vectors $v^{(1)}$, $v^{(2)}$, and $v^{(3)}$, and then using superposition. The first such system,

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 1 \end{bmatrix} \begin{Bmatrix} v_1^{(1)} \\ v_2^{(1)} \\ v_3^{(1)} \\ v_4^{(1)} \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}, \quad (2.5)$$

is solved by choosing $v_1^{(1)}$ arbitrarily and then determining the remaining terms using the subsequent equations in (2.5). Setting $v_1^{(1)} = a_{1,2}a_{2,3}$ leads to

$$\begin{aligned} v_2^{(1)} &= \frac{1}{a_{1,2}} - a_{1,1}a_{2,3}, \\ v_3^{(1)} &= -\frac{a_{2,2}}{a_{1,2}a_{2,3}} + a_{1,1}a_{2,2} - a_{1,2}a_{2,1}, \\ v_4^{(1)} &= \frac{a_{3,3}a_{2,2}}{a_{1,2}a_{2,3}} - \frac{a_{3,2}}{a_{1,2}} + a_{1,1}(a_{2,3}a_{3,2} - a_{2,2}a_{3,3}) + a_{1,2}(a_{2,1}a_{3,3} - a_{2,3}a_{3,1}). \end{aligned} \quad (2.6)$$

This process can be continued to compute all of the $v_j^{(k)}$ terms. The next system to be considered is:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 1 \end{bmatrix} \begin{Bmatrix} v_1^{(2)} \\ v_2^{(2)} \\ v_3^{(2)} \\ v_4^{(2)} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}. \quad (2.7)$$

Setting $v_1^{(2)} = a_{1,2}a_{2,3}$ leads to

$$\begin{aligned} v_2^{(2)} &= -a_{1,1}a_{2,3}, \\ v_3^{(2)} &= \frac{1}{a_{2,3}} + a_{1,1}a_{2,2} - a_{1,2}a_{2,1}, \\ v_4^{(2)} &= -\frac{a_{3,3}}{a_{2,3}} + a_{1,1}(a_{2,3}a_{3,2} - a_{2,2}a_{3,3}) + a_{1,2}(a_{2,1}a_{3,3} - a_{2,3}a_{3,1}). \end{aligned} \quad (2.8)$$

The third system is:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 1 \end{bmatrix} \begin{Bmatrix} v_1^{(3)} \\ v_2^{(3)} \\ v_3^{(3)} \\ v_4^{(3)} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}. \quad (2.9)$$

Setting $v_1^{(3)} = a_{1,2}a_{2,3}$ leads to

$$\begin{aligned} v_2^{(3)} &= -a_{1,1}a_{2,3}, \\ v_3^{(3)} &= a_{1,1}a_{2,2} - a_{1,2}a_{2,1}, \\ v_4^{(3)} &= 1 + a_{1,1}(a_{2,3}a_{3,2} - a_{2,2}a_{3,3}) + a_{1,2}(a_{2,1}a_{3,3} - a_{2,3}a_{3,1}). \end{aligned} \quad (2.10)$$

Superposition of the vectors $v^{(1)}$, $v^{(2)}$, and $v^{(3)}$ leads to a solution for (2.2), that is,

$$\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = b_1 \begin{Bmatrix} v_1^{(1)} \\ v_2^{(1)} \\ v_3^{(1)} \\ v_4^{(1)} \end{Bmatrix} + b_2 \begin{Bmatrix} v_1^{(2)} \\ v_2^{(2)} \\ v_3^{(2)} \\ v_4^{(2)} \end{Bmatrix} + (b_3 + x_4) \begin{Bmatrix} v_1^{(3)} \\ v_2^{(3)} \\ v_3^{(3)} \\ v_4^{(3)} \end{Bmatrix}. \quad (2.11)$$

From (2.11), note that

$$x_4 = b_1 v_4^{(1)} + b_2 v_4^{(2)} + (b_3 + x_4) v_4^{(3)}. \quad (2.12)$$

Substituting into (2.12) for $v_4^{(1)}$, $v_4^{(2)}$, and $v_4^{(3)}$ (and performing some algebra) yields

$$x_4 = -b_1 - b_2 - b_3 + \frac{b_1}{\det(A)} \left(\frac{a_{3,3}a_{2,2}}{a_{1,2}a_{2,3}} - \frac{a_{3,2}}{a_{1,2}} \right) - \frac{b_2}{\det(A)} \frac{a_{3,3}}{a_{2,3}} + \frac{b_3}{\det(A)}, \quad (2.13)$$

where

$$\det(A) = a_{1,1}(a_{2,2}a_{3,3} - a_{2,3}a_{3,2}) + a_{1,2}(a_{2,3}a_{3,1} - a_{2,1}a_{3,3}). \quad (2.14)$$

Finally, substituting x_4 and $v^{(1)}$, $v^{(2)}$, and $v^{(3)}$ into (2.11) leads to the solution to (2.1), after some algebra:

$$\begin{aligned} x_1 &= \frac{b_1(a_{2,2}a_{3,3} - a_{2,3}a_{3,2}) - b_2a_{1,2}a_{3,3} + b_3a_{1,2}a_{2,3}}{\det(A)}, \\ x_2 &= \frac{b_1(a_{2,3}a_{3,1} - a_{2,1}a_{3,3}) + b_2a_{1,1}a_{3,3} - b_3a_{1,1}a_{2,3}}{\det(A)}, \\ x_3 &= \frac{b_1(a_{2,1}a_{3,2} - a_{2,2}a_{3,1}) + b_2(a_{1,2}a_{3,1} - a_{1,1}a_{3,2}) + b_3(a_{1,1}a_{2,2} - a_{2,1}a_{1,2})}{\det(A)}. \end{aligned} \quad (2.15)$$

The next section extends the process for a system having an arbitrary dimension n . However, few comments are first warranted. It can be inferred by inspection of (2.11) that computations are minimized when m , the number of nonzero right hand side terms, is small. For example, when $m = 1$, only two of the $v^{(k)}$ vectors need to be computed, regardless of n . (Only one vector needs to be computed if b_n is the only nonzero right hand side term.) The number of operations to compute a single vector $v^{(k)}$ is on the order of that for the forward substitution process used in LU decomposition, or an order of magnitude smaller than performing the LU decomposition process itself. Thus, the approach can be useful when $m \ll n$. However, even when $m \approx n$, the solution for the $v^{(k)}$ vectors can be performed on m parallel processors, since each $v^{(k)}$ can be determined simultaneously on a single processor without any communication needed with any of the other processors. Thus, the run time in such a parallel algorithm should approach that for computing just one of the $v^{(k)}$ vectors.

A considerable amount of recent work has involved attempts to parallelize the solutions of linear systems [5–7] (e.g., involving LU or QR decompositions). Typically, the degree of parallelism achieved is only partial and is highly dependent on the structure and sparseness of A and requires varying degrees of communication between processors. In contrast, this approach provides a framework for full parallelization with m processors without any communication required between processors.

3. Proof of the General Case

This section provides a proof that the algorithm from the previous section is valid for the general case, that is, a lower Hessenberg linear system having arbitrary dimension n . Consider the system of equations

$$\sum_{j=1}^n a_{i,j} x_j = b_i, \quad (3.1)$$

for each value of i where $1 \leq i \leq n$, and $a_{i,j} = 0$ when $j > i + 1$. Following the procedure in the last section, consider the extended system

$$\sum_{j=1}^{i+1} a_{i,j} x_j = b_i + \delta_{i,n} x_{n+1}, \quad (3.2)$$

for each value of i , where $1 \leq i \leq n$. Here $a_{i,j} = 0$ when $j > i + 1$, $\delta_{i,n}$ is the Kronecker delta, $a_{n,n+1} = 1$, and $a_{i,i+1} \neq 0$ for all i . Consider further the n related systems of the form

$$\sum_{j=1}^{i+1} a_{i,j} v_j^{(k)} = \delta_{k,i}, \quad (3.3)$$

for each k , where $1 \leq k \leq n$, and for each i , where $1 \leq i \leq n$.

For a given k , each successive term of $v^{(k)}$ can thus be obtained by solving a single equation of the system (3.3) for one unknown (similar to the forward substitution process used in LU decomposition).

The $n + 1$ solutions $v^{(k)}$ for $k = 1, \dots, n$, once obtained, can then be superimposed as follows for $i \leq n + 1$:

$$x_i = \sum_{k=1}^n b_k v_i^{(k)} + x_{n+1} v_i^{(n)}. \quad (3.4)$$

Equation (3.4) satisfies (3.2) and provides an equation to determine x_{n+1} , that is,

$$x_{n+1} = \sum_{k=1}^n b_k v_{n+1}^{(k)} + x_{n+1} v_{n+1}^{(n)}. \quad (3.5)$$

Since (3.4) satisfies (3.2) and b_1, b_2, \dots, b_n are arbitrary, it follows that (3.5) leads to a solution for x_{n+1} . Substituting x_{n+1} obtained from (3.5) into (3.4) then determines all of the x_i terms. Finally, the n th equation of (3.2) is

$$\sum_{j=1}^n a_{n,j} x_j + x_{n+1} = b_n + x_{n+1}, \quad (3.6)$$

or

$$\sum_{j=1}^n a_{n,j} x_j = b_n, \quad (3.7)$$

which is identical to the n th equation in (3.1). Thus, this approach solves (3.2) and also solves (3.1), since (3.1) is a subset of (3.2) and is further isolated from the additional variable x_{n+1} .

4. Concluding Remarks

It has been shown that lower Hessenberg linear systems can be solved by considering related extended systems, first by directly solving a system with $n = 3$, and then extending the procedure to arbitrary n . The approach involving the development of m solution vectors from an extended system (having an additional column in the coefficient matrix and an additional introduced unknown) is highly parallelizable on m processors.

It should be noted that this approach also lends itself to lower q -Hessenberg linear systems, that is, involving square matrices $A = (a_{i,j})$, $i, j = 1, \dots, n$ such that $a_{ij} = 0$ when $j > i + q$. For example, $q = 2$ leads to two additional columns and two additional introduced variables x_{n+1} and x_{n+2} , that are solved with two additional equations. The proof is similar.

There is no limit in principle to the number of diagonals that can be added. Each will lead to an additional introduced unknown and an additional column to A . The process can thus be extended to many systems of practical interest, providing a framework for parallelization of such systems.

References

- [1] M. Elouafi and A. D. Aiat Hadj, "A new recursive algorithm for inverting Hessenberg matrices," *Applied Mathematics and Computation*, vol. 214, no. 2, pp. 497–499, 2009.
- [2] Y. Ikebe, "On inverses of Hessenberg matrices," *Linear Algebra and its Applications*, vol. 24, pp. 93–97, 1979.
- [3] M. J. Piff, "Inverses of banded and κ -Hessenberg matrices," *Linear Algebra and its Applications*, vol. 85, pp. 9–15, 1987.
- [4] X. Zhong, "On inverses and generalized inverses of Hessenberg matrices," *Linear Algebra and its Applications*, vol. 101, pp. 167–180, 1988.
- [5] J. W. Demmel, M. T. Heath, and H. A. van der Vorst, "Parallel numerical linear algebra," *Acta Numerica*, vol. 2, pp. 111–197, 1993.
- [6] G. A. Geist and C. H. Romine, "LU factorization algorithms on distributed-memory multiprocessor architectures," *Society for Industrial and Applied Mathematics*, vol. 9, no. 4, pp. 639–649, 1988.
- [7] I. E. Venetis and G. R. Gao, "Mapping the LU decomposition on a many-core architecture: challenges and solutions," in *Proceedings of the 6th ACM Conference on Computing Frontiers (CF '09)*, pp. 71–80, Ischia, Italy, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

