Chapter HNM
Householder Numerically with *Mathematica*

By Robert Hildebrand

Chapter HNM
Householder Numerically with *Mathematica*

This chapter written by Robert Hildebrand

Definition TDM
Tridiagonal Matrix

The $n \times n$ square matrix A is **tridiagonal** provided that $[A]_{ij} = 0$ whenever $i \geq j + 2$ and whenever $j \geq i + 2$.

△

Example TDMS4
Tridiagonal Matrix Size 4

$$A = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 3 & 4 & 5 & 0 \\ 0 & 6 & 7 & 8 \\ 0 & 0 & 9 & 10 \end{bmatrix}$$

is a tridiagonal matrix of size 4.

⊠

In this section, we will use *householder transformations* to reduce symmetric matrix A to a similar, symmetric, tridiagonal matrix A_{n-1}. The subscript will make sense later. A tridiagonal matrix is an important starting point to applying numerical techniques of *QR factorization*, which would be the section following this one if it existed. Since A_{n-1} will be similar to A, Theorem SMEE guarantees that they will have equal eigenvalues. This preservation of eigenvalues is essential for QR methods.

Note: For the purposes of this section, we will restrain our work to $\mathbb{R}$, and not venture into $\mathbb{C}$.

Definition HM
Householder Matrix

Let $\mathbf{x} \in \mathbb{R}^n$. Then the $n \times n$ matrix

$$H = I_n - \frac{2\mathbf{x}\mathbf{x}^t}{||\mathbf{x}||^2}$$

is a **Householder Matrix**.

△

Notice that the identity matrix is not a householder matrix as it would require $\mathbf{x}$ to be the zero vector, which then would cause a division by zero. We will later use $\mathbf{x}$'s with norm 1, which will reduce the above definition to

$$H = I_n - 2\mathbf{x}\mathbf{x}^t$$

Theorem HMSU
Householder Matrix is Symmetric and Unitary

Let $\mathbf{x} \in \mathbb{R}^n$ and let H be the $n \times n$ householder matrix given by

$$H = I_n - \frac{2\mathbf{x}\mathbf{x}^t}{||\mathbf{x}||^2}.$$

Then H is symmetric and unitary.

□

Proof. We will begin by showing that H is symmetric.

$$\begin{aligned}
H^t &= \left(I_n - \frac{2\mathbf{x}\mathbf{x}^t}{\|\mathbf{x}\|^2}\right)^t \\
&= (I_n)^t - \left(\frac{2\mathbf{x}\mathbf{x}^t}{\|\mathbf{x}\|^2}\right)^t && \text{Theorem TMA} \\
&= I_n - \frac{2}{\|\mathbf{x}\|^2}\left(\mathbf{x}\mathbf{x}^t\right)^t && \text{Definition SYM, Theorem TMSM} \\
&= I_n - \frac{2}{\|\mathbf{x}\|^2}(\mathbf{x}^t)^t\mathbf{x}^t && \text{Theorem MMT} \\
&= I_n - \frac{2}{\|\mathbf{x}\|^2}\mathbf{x}\mathbf{x}^t && \text{Theorem TT} \\
&= H
\end{aligned}$$

We now use the symmetry to show that H is unitary. Remember that we are only working in $\mathbb{R}$, so $H^* = H^t$.

$$\begin{aligned}
H^tH = HH &= \left(I_n - \frac{2\mathbf{x}\mathbf{x}^t}{\|\mathbf{x}\|^2}\right)\left(I_n - \frac{2\mathbf{x}\mathbf{x}^t}{\|\mathbf{x}\|^2}\right) \\
&= I_nI_n - 4I_n\frac{\mathbf{x}\mathbf{x}^t}{\|\mathbf{x}\|^2} - 4\frac{\mathbf{x}\mathbf{x}^t\mathbf{x}\mathbf{x}^t}{\|\mathbf{x}\|^2\|\mathbf{x}\|^2} && \text{Theorem MMDAA} \\
&= I_n - 4\frac{\mathbf{x}\mathbf{x}^t}{\|\mathbf{x}\|^2} - 4\frac{\mathbf{x}\,\|\mathbf{x}\|^2\,\mathbf{x}^t}{\|\mathbf{x}\|^2\|\mathbf{x}\|^2} && \text{Theorem IPN} \\
&= I_n
\end{aligned}$$

■

Section HMRSM
Householder Method to Reduce Symmetric Matrices

We will use an iterative process with householder matrices to tridiagonalize a symmetric matrix A of size n. We will find the appropriate householder transformation to operate on the first column of A, apply the transformation, find the next householder transformation needed to operate on the second column and so forth. This will result in a series of $n-2$ householder transformations. If we let $A_1 = A$, and apply $n-2$ transformations to A_1, then the result will be A_{n-1}. We will begin with a lemma to help us along.

Lemma SMPS
Symmetric Matrices Preserve Symmetry

Let A and B be symmetric matrices of size n. Then BAB is also a symmetric matrix of size n.

□

Proof.

$$\begin{aligned}
(BAB)^t &= B^t(BA)^t && \text{Theorem MMT} \\
&= B^tA^tB^t && \text{Theorem MMT} \\
&= BAB && \text{Definition SYM for A and B}
\end{aligned}$$

By definition, BAB is symmetric. ■

Theorem HMRSM
Householder Method to Reduce Symmetric Matrices

Let A be a symmetric matrix of size n. Then A is similar to a symmetric, tridiagonal matrix B.

□

Proof. Let H be the householder transformation based on the column vector $\mathbf{x}$. Let

$$A = [\mathbf{a}_1|\mathbf{a}_2|...|\mathbf{a}_n] \qquad \text{and} \qquad B = [\mathbf{b}_1|\mathbf{b}_2|...|\mathbf{b}_n].$$

We will apply a series of householder transformations to A in order to create B. Since B is tridiagonal, $[\mathbf{b}_j] = 0$ if $i \geq j+2$.

$$H_{n-2}H_{n-1}...H_2H_1AH_1H_2...H_{n-1}H_{n-2}$$

By iterating Lemma SMPS $n-2$ times, we find that the result of the above multiplication is symmetric. The goal of H_j is to reduce the j^{th} column of A, $\mathbf{a}_j$, to $\mathbf{b}_j$, where

$$\begin{aligned}
[\mathbf{b}_j]_i &= [\mathbf{a}_j]_i && \text{for } i \leq j \\
[\mathbf{b}_j]_{j+1} &= \alpha && \\
[\mathbf{b}_j]_i &= 0 && \text{for } i \geq j+2
\end{aligned}$$

That is, we want

$$H\mathbf{a}_j = \mathbf{b}_j = \begin{bmatrix} [\mathbf{a}_j]_1 \\ [\mathbf{a}_j]_2 \\ \vdots \\ [\mathbf{a}_j]_j \\ \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

This relationship comes out of the definition of matrix multiplication. The value of α is unknown at this point.

Notice that we are leaving the first j entries unaltered. This is foreshadowing that the $j \times j$ matrix formed by taking the first j rows and j columns to be the identity matrix of size j. To preserve symmetry and eigenvectors (Theorem SMEE), we will multiply on the right by H. Notice that

$$\mathbf{a}_j^t H = \mathbf{a}_j^t H^t = (H\mathbf{a}_j)^t = \mathbf{b}_j^t.$$

This shows that multiplying from the right with H will give us the desired zeros that lie above the diagonal. Once we have created H to reduce the j^{th} column and row, then we just iterate this process; solve for H and reduce the first column and row, resolve for H and reduce the second column and row, etc.

Now we need to show that such a reduction can happen. We must find $\mathbf{x}$ that makes $H\mathbf{a}_j = \mathbf{b}_j$ since H is dependent on $\mathbf{x}$. The two freedoms that give ourselves are α, which we will discover its value later, and the norm of $\mathbf{x}$ which we choose to be 1, that is $||\mathbf{x}|| = 1$. We start by finding a relation between the entries of $\mathbf{x}$ and the entries of $\mathbf{b}_j$ based on the desired results described above.

Let $r_j = \mathbf{x}^t\mathbf{a}_j$. We start out work entry by entry.

$$\begin{aligned}
[\mathbf{b}_j]_i &= [H\mathbf{a}_j]_i && \text{Definition MM}\\
&= \sum_{k=1}^{n}[H]_{ik}[\mathbf{a}_j]_k && \text{Theorem EMP}\\
&= \sum_{k=1}^{n}[I_n - 2\mathbf{x}\mathbf{x}^t]_{ik}[\mathbf{a}_j]_k && \text{Definition HM}\\
&= \sum_{k=1}^{n}([I_n]_{ik} - [2\mathbf{x}\mathbf{x}^t]_{ik})[\mathbf{a}_j]_k && \text{Definition MA}\\
&= \sum_{k=1}^{n}([I_n]_{ik}[\mathbf{a}_j]_k - [2\mathbf{x}\mathbf{x}^t]_{ik}[\mathbf{a}_j]_k) && \text{Distributivity in } \mathbb{C}\\
&= \sum_{k=1}^{n}[I_n]_{ik}[\mathbf{a}_j]_k - \sum_{k=1}^{n}[2\mathbf{x}\mathbf{x}^t]_{ik}[\mathbf{a}_j]_k && \text{Commutativity in } \mathbb{C}\\
&= [\mathbf{a}_j]_i - [2\mathbf{x}\mathbf{x}^t\mathbf{a}_j]_i && \text{Theorem MMIM, Theorem EMP}\\
&= [\mathbf{a}_j]_i - [2(r_j)\mathbf{x}]_i && \text{Theorem MMA}\\
&= [\mathbf{a}_j - 2r_j\mathbf{x}]_i && \text{Definition CVA}
\end{aligned}$$

$$[\mathbf{x}]_i = 0 \qquad \text{for } i \leq j \tag{1}$$
$$2r_j[\mathbf{x}]_i = [\mathbf{a}_j]_i - \alpha \qquad \text{for } i = j+1 \tag{2}$$
$$2r_j[\mathbf{x}]_i = [\mathbf{a}_j]_i \qquad \text{for } i \geq j+2 \tag{3}$$

We now have an equation for every entry of $\mathbf{x}$. Our last task is to choose values for r_j and α. Forget that $r_j = x^t\mathbf{a}_j$ and now just think of r_j as a constant. Using the above relationships, we without motivation multiply equation (1) by $2r_j$, square each of the equations (1), (2), and (3) and then sum the equations for all indices $1 \leq j \leq n$.

$$\begin{aligned}
& \sum_{k=1}^{j}(2r_j[\mathbf{x}]_k)^2 &&= \sum_{k=1}^{j}0^2\\
+ \quad & (2r_j[\mathbf{x}]_{(j+1)})^2 &&= ([\mathbf{a}_j]_{(j+1)} - \alpha)^2\\
+ \quad & \sum_{k=j+2}^{n}([2r_j\mathbf{x}]_k)^2 &&= \sum_{k=j+2}^{n}([\mathbf{a}_j]_k)^2\\
\Rightarrow \quad & 4r_j^2\sum_{k=1}^{n}[\mathbf{x}]_i^2 &&= \alpha^2 - 2\alpha[\mathbf{a}_j]_{(j+1)} + \sum_{k=j+1}^{n}([\mathbf{a}_j]_k)^2
\end{aligned}$$

$$\Leftrightarrow \quad 4r_j^2 = \alpha^2 - 2\alpha[\mathbf{a}_j]_{(j+1)} + \sum_{k=j+1}^{n}([\mathbf{a}_j]_k)^2 \qquad \text{Since we choose } ||\mathbf{x}|| = 1 \tag{4}$$

This is not yet enough information to determine α and r_j, so we will derive another realtionship for alpha.

$$\sum_{k=1}^{j}[\mathbf{a}_j]_k^2 + \alpha^2 = \begin{bmatrix}[\mathbf{a}_j]_1 & [\mathbf{a}_j]_2 & \dots & [\mathbf{a}_j]_{j1} & \alpha & 0 & \dots & 0\end{bmatrix} \begin{bmatrix}[\mathbf{a}_j]_1 \\ [\mathbf{a}_j]_2 \\ \vdots \\ [\mathbf{a}_j]_j \\ \alpha \\ 0 \\ \vdots \\ 0\end{bmatrix}$$

$$\begin{aligned}
&= \langle \mathbf{b}, \mathbf{b} \rangle && \text{Definition of } \mathbf{b} \\
&= \langle H\mathbf{a}_j, H\mathbf{a}_j \rangle && \text{Definition of } \mathbf{b} \\
&= \left\langle H^t H\mathbf{a}_j, \mathbf{a}_j \right\rangle && \text{Theorem AIP} \\
&= \langle \mathbf{a}_j, \mathbf{a}_j \rangle && \text{Theorem HMSU} \\
&= \sum_{k=1}^{n}[\mathbf{a}_j]_k^2 && \text{Definition IP}
\end{aligned}$$

$$\Rightarrow \alpha^2 = \sum_{k=j+1}^{n}[\mathbf{a}_j]_k^2$$

We can choose α to be positive or negative. Since our choice of α will dictate the value of r_j through (4), we choose α to have the opposite sign as $[\mathbf{a}_j]_{j+1}$. This is done to reduce round off error when computing r_j. Round off error becomes an issue when subtracting two numbers that are nearly equal, so we are making this decision to avoid such a dilemma. Thus we choose

$$c = 1 \qquad \text{if } [\mathbf{a}_j]_{j+1} \geq 0 \qquad c = 2 \text{ if } [\mathbf{a}_j]_{j+1} < 0$$

$$\begin{aligned}
\alpha &= (-1)^c \left(\sum_{k=j+1}^{n} [\mathbf{a}_j]_{k1}^2 \right)^{1/2} \\
\Rightarrow \quad 4r_j^2 &= \alpha^2 - 2\alpha[\mathbf{a}_j]_{(j+1)} + \sum_{k=j+1}^{n} ([\mathbf{a}_j]_{k1})^2 \\
&= \alpha^2 - 2\alpha[\mathbf{a}_j]_{(j+1)} + \alpha^2 \\
&= 2\alpha^2 - 2\alpha[\mathbf{a}_j]_{(j+1)} \\
\Rightarrow \quad r_j &= \left(\tfrac{1}{2}\alpha^2 - \tfrac{1}{2}[\mathbf{a}_j]_{j+1}\alpha\right)^{1/2}
\end{aligned}$$

Finally, we can completely solve for $\mathbf{x}$ based on equations (1), (2), and (3). I leave it as an exercise to double check that $||\mathbf{x}|| = 1$. ■

Now what? Now we implement this tool!

Example FTHT
First Try with Householder Tridiagonalization

$$A = \begin{bmatrix} 4 & 2 & -2 & 1 \\ 2 & 3 & 2 & 1 \\ -2 & 2 & 1 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}$$

To tridiagonalize we apply the details of the proof for Theorem HMRSM and start by solving for α, r_j and $\mathbf{x}$ with $j = 1$.

$$\alpha = (-1)^1\sqrt{2^2 + (-2)^2 + 1^2} = -3 \qquad r_1 = \left(\tfrac{1}{2}(-3)^2 - \tfrac{1}{2}2(-3)\right)^{1/2} = \sqrt{15/2}$$

$$\mathbf{x}_1 = \begin{bmatrix} 0 \\ \sqrt{\frac{5}{6}} \\ -\sqrt{\frac{2}{15}} \\ \frac{1}{\sqrt{30}} \end{bmatrix} \Rightarrow H_1 = I_n - 2\mathbf{x}_1\mathbf{x}_1^t = I_4 - \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{5}{3} & -\frac{2}{3} & \frac{1}{3} \\ 0 & -\frac{2}{3} & \frac{4}{15} & -\frac{2}{15} \\ 0 & \frac{1}{3} & -\frac{2}{15} & \frac{1}{15} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{2}{3} & \frac{2}{3} & -\frac{1}{3} \\ 0 & \frac{2}{3} & \frac{11}{15} & \frac{2}{15} \\ 0 & -\frac{1}{3} & \frac{2}{15} & \frac{14}{15} \end{bmatrix}$$

$$A_2 = H_1A_1H_1 = \begin{bmatrix} 4 & -3 & 0 & 0 \\ -3 & \frac{2}{3} & -\frac{4}{3} & -1 \\ 0 & -\frac{4}{3} & \frac{101}{25} & -\frac{4}{75} \\ 0 & -1 & -\frac{4}{75} & \frac{97}{75} \end{bmatrix}$$

Now we iterate the process a second time.

$$\alpha = (-1)^2\sqrt{\left(\tfrac{-4}{3}\right)^2 + (-1)^2} = \sqrt{\tfrac{16}{9} + 1} = \tfrac{5}{3} \qquad r_2 = \left(\tfrac{1}{2}\left(\tfrac{5}{3}\right)^2 - \tfrac{1}{2}\tfrac{-4}{3}\tfrac{5}{3}\right)^{1/2} = \left(\tfrac{25}{18} + \tfrac{20}{18}\right)^{1/2} = \sqrt{\tfrac{5}{2}}$$

$$\mathbf{x}_2 = \begin{bmatrix} 0 \\ 0 \\ -\frac{3}{\sqrt{10}} \\ -\frac{1}{\sqrt{10}} \end{bmatrix} \Rightarrow \qquad H_2 = I_4 - 2\mathbf{x}_2\mathbf{x}_2^t = I_4 - \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{9}{5} & \frac{3}{5} \\ 0 & 0 & \frac{3}{5} & \frac{1}{5} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{4}{5} & -\frac{3}{5} \\ 0 & 0 & -\frac{3}{5} & \frac{4}{5} \end{bmatrix}$$

$$A_3 = H_2A_2H_2 = \begin{bmatrix} 4 & -3 & 0 & 0 \\ -3 & \frac{2}{3} & \frac{5}{3} & 0 \\ 0 & \frac{5}{3} & 3 & \frac{4}{3} \\ 0 & 0 & \frac{4}{3} & \frac{7}{3} \end{bmatrix}$$

$\boxtimes$

As you can see, there are many calculations that must be done, which makes this technique very messy to be done by hand.

Example MIHM
***Mathematica* Implementation of Householder's Method**

Mathematica is not only a handy graphing calculator, it is also a programming language. *Mathematica* is particularly useful for programming when doing matrix operations because it has those algorithims already built in. The following is a program that will implement householder's method using *Mathematica*.
First, we must choose a symmetric matrix of size n. Below is a sample matrix of size 4. We input this one into *Mathematica*. Before diving into the code, note that we can reference the i^{th} row of A in *Mathematica* by A[[i]]. And we can reference the j^{th} entry of the i^{th} row by A[$[i,j]$] (ex. A[$[1,1]$] $= 4$).

$$\mathtt{A} = \begin{pmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{pmatrix};$$

We need to initialize our variables and input any starting conditions which we might need later. To make things easier later, we have built a column vector of size n with all zero's and called it zeroVector. The code *zeroVector* = { } initializes the variable and the *For loop* builds the vector with *Append* and stores zeros into its entries. We have also defined *Hlist* and *Alist* which hold list of the householder transformations and the transformed A matrix respectively for each step; basically the list will save our work.

```
n = Length[A[[1]]];
zeroVector = {};
    For[i = 1, i ≤ n,
      zeroVector = Append[zeroVector, {0}];
      i++
      ];
Alist = {A};
Hlist = {};
```

At the beginning of our algorithm, we start a *For loop* to encompass the everything, and then solve for α and r_j for the current iteration. Notice that we use an *If* statement to determine the sign of $A[\,[j+1, j]\,]$. There is a function *Sign*[ ] in *Mathematica*, but this will assign a value of 0 if the input is 0. If we had used *Mathematica*'s *Sign*[ ] we would have the possibility of assigning $\alpha = 0 \Rightarrow r_j = 0$, which would make the algorithm divide by zero.

```
For[j = 1, j ≤ n - 2,

  If[A[[j + 1, j]] ≥ 0, c = 1, c = 2];
```

$$\alpha = (-1)^{c} \left(\sum_{k=j+1}^{n} A[[k, j]]^2 \right)^{1/2};$$

$$r = \left(\frac{1}{2} \alpha^2 - \frac{1}{2} \alpha A[[j+1, j]] \right)^{1/2};$$

After computing α and r_j, we then solve for $\mathbf{x}_j$, where another *For loop* is necessary to compute for the entries $j+2$ to n. Notice that we make use of the *zeroVector* by first making $\mathbf{x}$ =*zeroVector*, and then overwrite the values for entries $j+1$ to n.

```
x = zeroVector;
```

$$x[[j+1, 1]] = \frac{A[[j+1, j]] - \alpha}{2 r};$$

```
For[k = j + 2, k ≤ n,
```

$$x[[k, 1]] = \frac{A[[k, j]]}{2 r};$$

```
 k++ ];
```

Finally, we compute H followed by computing HAH. We recycle the variables $H, A, \mathbf{x}, \alpha$, and r in each iteration, but we are interested in knowing which householder matrices were used and the transformed A_i along the way, so we store H and A into *Hlist* and *Alist* after each iteration. We end the code with the ending syntax of our *For loop*.

```
H = IdentityMatrix[n] - 2 x.Transpose[x];
A = H.A.H;
Hlist = Append[Hlist, H];
Alist = Append[Alist, A];

j++];
```

Now we can print our final result. There is also the option to print the work done, which for $n = 4$ would just be printing Hlist[[1]], Hlist[[2]], Alist[[1]], Alist[[2]], Alist[[3]].

```
Print[A // MatrixForm];
```

$$\begin{pmatrix} 4 & -3 & 0 & 0 \\ -3 & \frac{2}{3} & \frac{5}{3} & 0 \\ 0 & \frac{5}{3} & 3 & \frac{4}{3} \\ 0 & 0 & \frac{4}{3} & \frac{7}{3} \end{pmatrix}$$

⊠

The above code is correct, but slightly incomplete. *Mathematica* is a great tool and has the capability of exactly computing values such as

$$\frac{\sqrt{2}\sqrt{5}}{\sqrt{2}} = \sqrt{\frac{10}{3}}$$

Exactness is ideal, but is highly unefficient with regaurds to computing time. The code above will compute the exact solution, which is practical for a relatively small matrix (i.e. size 4), but not practical for larger matrices.

To remedy the code, warp the beginning matrix A with the numerical output function and redefine your matrix, A = N[A]. This will drastically speed up the processing time by changing the inputs to decimal approximations. This has one negative effect: round off error. The output will have very small differences from the exact computation, which realuts in relatively very small numbers where there should be zeros. For this reason, the output no longer appears tridiagonal. This error is developed because of the initial decimal approximations and that a computer can only hold a finite number of digits. The fix, for now, is to simply overwrite the should be zeros with zeros. This can be done with the following code.

```
For[i = 1, i ≤ n,
 For[j = i + 2, j ≤ n,
  A[[i, j]] = 0;
  A[[j, i]] = 0;
  j++];
 i++];
```

If you are not satisfied with this fix, I encourage you to take a course in Numerical Analysis to dive deeper into the topic.

Example CNAHM
Correcting Numerical Approximated Householder Method

$$A = \begin{bmatrix} -42 & 43 & -2 & 28 \\ 43 & -98 & 72 & -26 \\ -2 & 72 & -96 & 53 \\ 28 & -26 & 53 & 54 \end{bmatrix}$$

We implement the code given in Example MIHM and use the numerical approximations this time to obtain

$$A_3 = \begin{bmatrix} -42.0000 & -51.3517 & -1.9333 \times 10^{-15} & 2.9807 \times 10^{-15} \\ -51.3517 & -83.4956 & 107.2608 & -2.1316 \times 10^{-14} \\ -1.9339 \times 10^{-15} & 107.2608 & -45.7669 & -58.6633 \\ 2.9807 \times 10^{-15} & -1.421 \times 10^{-14} & -58.6633 & -10.7373 \end{bmatrix}$$

After applying the correcting code, we obtain

$$A_3 = \begin{bmatrix} -42.0000 & -51.3517 & 0 & 0 \\ -51.3517 & -83.4956 & 107.2608 & 0 \\ 0 & 107.2608 & -45.7669 & -58.6633 \\ 0 & 0 & -58.6633 & -10.7373 \end{bmatrix}$$

We can check our accuracy by having *Mathematica* numerically compute the eigenvalues of our original matrix A and the resulting matrix A_3. The eigenvalues for the original matrix are approximately

$$\{-191.73180785773600, 76.82569425480480, -58.02072265676360, -9.07316374030525\},$$

while the eigenvalues for the end matrix A_3 are approximately

$$\{-191.73180785773600, 76.82569425480480, -58.02072265676360, -9.07316374030523\}.$$

Notice that there is only one dighit that does not correspond, which is the last dighit given on the last eigenvalue. This is a specific example that helps to justify the correction process suggested above.

Concluding Remarks

The householder matrix is structured to easily solve for the transformation necessary to obtain a desired output. Householder matrices have wider application than just tridiagonalizing a symmetric matrix. The next step would be a QR factorization, which also needs householder's matrix. QR only multiplies H from the left hand side. For example we might compute $H_4H_3H_2H_1A = R$. QR factorization is a very accurate technique for numerically computing eigenvalues because it minimizes round off error. The numerical QR method relies on a starting with a symmetric, tridiagonal matrix. If the starting point is only a symmetric matrix, then householder's method must first be applied to tridiagonalize the matrix. For more information on QR factorization, or to learn more about householder matrices, please reference the texts in the bibliography.

Bibliography

Burden, Richard L., and J. Douglas Faires. Numerical Analysis. 8th Edition. Belmont: Thomson Books/Cole, 2005.

Gill, Phillip E., Walter Murray, and Margaret H. Wright. Numerical Linear Algebra and Optimization. New York: Addison-Wesely, 1991.

Kerl, John. "The Householder Transformation in Numerical Analysis." January 22, 2007

Stewart, G.W.. Matrix Algorithms Volume II: Eigensystems. Philadelphia: Siam, 2001.