# *Lesson Plan 7 on Orthogonal Matrices, Householder Reflections, QR Matrix Decomposition and Orthonormal Bases*

Frank Uhlig

*Department of Mathematics and Statistics,*
*Auburn University, Auburn, AL 36849-5310, USA,*
**uhligfd@auburn.edu**

Lesson Plan 7 interprets Gaussian elimination of a matrix $A_{n,m}$ as an LR factorization and extends matrix factorizations to orthogonal elimination matrices $Q$ in the QR factorization. It introduces orthogonal matrices $Q_{n,n}$ with orthonormal columns so that $Q^TQ = I_n$ and then specifically studies Householder matrices $H = I_n - 2u\,u^T$ for unit vectors $u$. Householders are applied to solve unsolvable linear equations as best as one can, to find ONBs of subspaces and to formalize eigenvalue deflation for Krylov vector iterations, shoring up the intuitive methods for finding matrix eigendata via Krylov from the end of Lesson Plan 5.

(version August 6, 2023) iii + 24 + VIII p.; 4 - 6 class days

# Concepts, Notions and Definitions in Lesson Plan 7

Orthonormal Vectors

Vector Dot Product

Transposed Matrix

LR Factorization

Upper Triangular Matrix

Orthogonal Matrix

Dyad

Orthogonal Elimination Matrix

(continued)

Householder Transformation

Eigenvalues and Eigenvectors

Nullspace

Matlab Validation

Orthonormal Basis

Least Squares

Normal Equation

Condition of Matrices

Extended Krylov Eigendata Method

*In this lesson we explore orthogonal unit vectors $v_i \in \mathbb{R}^n$ (or $\mathbb{C}^n$) that play a special role in Matrix Theory and Computations.*

*Matrices $Q \in \mathbb{R}_{n,n}$ with mutually orthogonal unit vectors as columns or rows are called orthogonal matrices. Orthogonal matrices $Q_{n,n}$ are defined by the equation $Q^T \cdot Q = I_n$ where the transposed matrix $Q^T$ contains the column entries of $Q$, but written row wise.*

*We study Householder Transformations $I - 2u\,u^T$ as orthogonal elimination matrices and introduce the QR decomposition of matrices $A_{n,m} = Q_{n,n} \cdot R_{n,m}$.*

*We apply QR factorizations to solve unsolvable linear equations and to compute orthonormal bases for subspaces.*

*Numerical codes are built to validate the theoretical results.*

*An epilogue then closes this matrix based linear algebra course.*

If $\{u_i\}$ are $k$ mutually orthogonal unit vectors in $\mathbb{R}^n$, then for each $i \neq j$ the vector dot products $u_i^T \cdot u_j = 0$ due to the mutual orthogonality and if $i = j$ then $u_i^T \cdot u_j = 1$ since each $u_i$ is a unit vector.

Let $U_{n,k} = \begin{pmatrix} \vdots & & \vdots \\ u_1 & \cdots & u_k \\ \vdots & & \vdots \end{pmatrix}$ be the column vector matrix for the $u_i$. Then the transposed matrix $U_{k,n}^T = \begin{pmatrix} \cdots & u_1^T & \cdots \\ & \vdots & \\ \cdots & u_k^T & \cdots \end{pmatrix}$ is the row vector matrix of the $u_i$.

And the matrix product $\mathbf{U_{k,n}^T} \cdot \mathbf{U_{n,k}}$ is the identity matrix $I_k$, with ones on the diagonal due to the normalized $u_i$ and zeros everywhere else due to their mutual orthogonality.

*Clearly* $k \leq n$ *here. Why?*

*Square* **orthogonal matrices** $Q_{n,n}$, *defined by* $Q^TQ = I_n$, *have low computational costs and errors and they are of great help in* **modern matrix computations**.

Recall the LR factorization of matrices $A_{n,m}$ by using a sequence of Gaussian elimination matrices $G_k$.
A row-updated matrix $\tilde{A}$ with pivot $\boxed{1}$ in position $(k,k)$ asks for an elimination matrix $G_k$ that zeros out all entries in column $k$ below position $k$. This $G_k$ has the simple form below :

$$\tilde{A} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & * & & \\ & & 1 & & & & \\ & & & \boxed{1} & * & \cdots & * \\ & & & \vdots & * & \cdots & * \\ & 0 & & \tilde{a}_k & \vdots & \ddots & \vdots \\ & & & \vdots & * & \cdots & * \end{pmatrix}, G_k = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & 0 & \\ & & 1 & & & & \\ & & & \boxed{1} & & & \\ & & & \vdots & 1 & & \\ & 0 & & -\tilde{a}_k & 0 & \ddots & \\ & & & \vdots & 0 & 0 & 1 \end{pmatrix}$$

Each Gaussian elimination matrix $G_k$ for $k \leq n-1$ and $A_{n,n}$ is the sum of $I_n$ plus a rank 1 lower triangular matrix.

We want to repeat this low rank perturbation of $I$ method with orthogonal matrices $Q_k$ to write $\mathbf{A_{n,m} = Q_{n,n} \cdot R_{n,m}}$ where $Q^T \cdot Q = I_n$ and $R$ is upper triangular.

For a unit vector $u \in \mathbb{R}^n$, the matrix

$$H = I_n - 2u \cdot u^T$$

is a rank 1 perturbation of $I_n$ and orthogonal since

$$\begin{aligned} H^T \cdot H &= (I - 2u\, u^T)^T \cdot (I - 2u\, u^T) = (I - 2u\, u^T)^2 \\ &= I_n - 4\, u\, u^T + 4\, u\, u^T \cdot u\, u^T = I_n \qquad (*) \end{aligned}$$

and $u^T \cdot u = 1$.

How do we choose $u$ so that the $n-k+1$ entries $\begin{pmatrix} \vdots \\ a_k \\ \vdots \end{pmatrix}_{n-k+1,k}$ in column $k$ of the updated partially upper triangular matrix $\tilde{A}_k$ are zeroed out by $H = I - 2u\, u^T$.

Gaussian elimination matrices $G_k$ for column $k$ are the identity matrix $I_n$ plus a non-zero fill-in from position $k+1$ on down in column $k$.

Orthogonal elimination matrices $H_k$ for column $k$ of $\tilde{A}$ are block-diagonal with the identity matrix $I_{k-1}$ on top and entries from row and column $k$ on down and to the right in an orthogonal block $U_{n-k+1,n-k+1}$, i.e.,

$$H_k = \texttt{blkdiag}(I_{k-1}, U)$$

in Matlab notation.

The orthogonal elimination block $U = I_{n-k+1} - 2v \cdot v^T$ of Householder type must map $v = \begin{pmatrix} \vdots \\ a_k \\ \vdots \end{pmatrix}_{n-k+1}$ to $\begin{pmatrix} \pm\|v\| \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{n-k+1}$ to eliminate the entries in column $k$ of $\tilde{A}$ below the $\tilde{A}$'s diagonal and carry the length of $v$ on the updated $k$ th diagonal spot.

For any nonzero column vector $v \in \mathbb{R}^\ell$, the column times row vector product $\mathbf{v}_{\ell,\mathbf{1}} \cdot \mathbf{v}^{\mathbf{T}}_{\mathbf{1},\ell}$ is an $\ell$ by $\ell$ dense matrix of rank 1 as all its rows are multiples of $v^T$ and its columns multiples of $v$. Such $v_\ell \cdot v_\ell^T$ $\ell$ by $\ell$ matrix products are called dyads.

Note that not all rank 1 matrices are dyads.

*Find a matrix of rank 1 that cannot be expressed as a dyad.*

We construct dyad defined orthogonal elimination matrices $Q_{n,n} = I_n - 2v \cdot v^T$ most simply from their eigenstructures.

**Lemma 1** : *If* $Q^T Q = I_n$, *then for all* $x \in \mathbb{R}^n$, $\|Qx\| = \|x\|$.

**Lemma 2** : *If* $Q^T Q = I_n$, *then all eigenvalues* $\lambda$ *of* $Q$ *lie on the unit circle of the complex plane or* $|\lambda| = 1$.

**Lemma 3** : *For any unit column vector* $x \in \mathbb{R}^n$ *the matrix* $Q = I_n - \alpha\, x\, x^T$ *is orthogonal if and only if* $\alpha = 2$ *or* $\alpha = 0$.

**Proofs :**

(1) From the euclidean norm definition

$$\begin{aligned} \|Qx\|^2 &= (Qx)^T Qx = x^T\, Q^T Q\, x = x^T\, (Q^T\, Q)\, x \\ &= x^T I_n\, x = x^T\, x = \|x\|^2. \end{aligned}$$

(2) If $Qy = \lambda y$ then $\|Qy\| = \|\lambda y\| = |\lambda| \, \|y\|$. From Lemma 1 we know that $\|Qy\| = \|y\|$. This implies that $|\lambda| = 1$.

(3) Use the earlier formula $(*)$ with 2 replaced by $\alpha$ to obtain

$$\begin{aligned} Q^T \cdot Q &= (I_n - \alpha \, x \, x^T)^T \cdot (I_n - \alpha \, x \, x^T) \\ &= I_n - 2\alpha \, x \, x^T + \alpha^2 \, x \, x^T \stackrel{*}{=} I_n \end{aligned}$$

since $x \, x^T \cdot x \, x^T = x \, x^T$ as $x^T \, x = 0$. Thus the latter equal sign $*$ is true exactly when $2\alpha = \alpha^2$, or if and only if $\alpha = 2$ or $\alpha = 0$. □

What are the **eigenvalues and eigenvectors of the orthogonal matrix** $Q_{n,n} = I_n - 2 \, u \, u^T$ for any given unit vector $u$?

*Determinant and polynomial root-finder eigenvalue methods of old cannot help here at all. $Q$ is a rank 1 perturbation of the identity map. $Q$ is orthogonal, all its eigenvalues have absolute value 1. What may the eigenvectors be?*

*What happens to $u$ when it is mapped by $Q$?*

$$\mathbf{Q\,u = (I - 2\,u\,u^T)u = u - 2\,u\,u^T\,u = u - 2\,u = -\,u}$$

since $u^T \cdot u = 1$. Clearly $u$ is an eigenvector of $Q$ for the eigenvalue $\lambda_1 = -1$.

When $u$ is the defining unit vector for $Q$, then $Q = I - 2uu^T$ reverses the direction of $u$. If in the $k$th elimination we choose $u$ as the vector from $\|v_k\|e_k$ to the $k$th lower triangular column $v_k$ of $A$ then $Q = I - 2uu^T$'s direction reversal property sends $v_k$ to $\|v_k\|e_k$ as desired for lower triangular elimination of $A$.

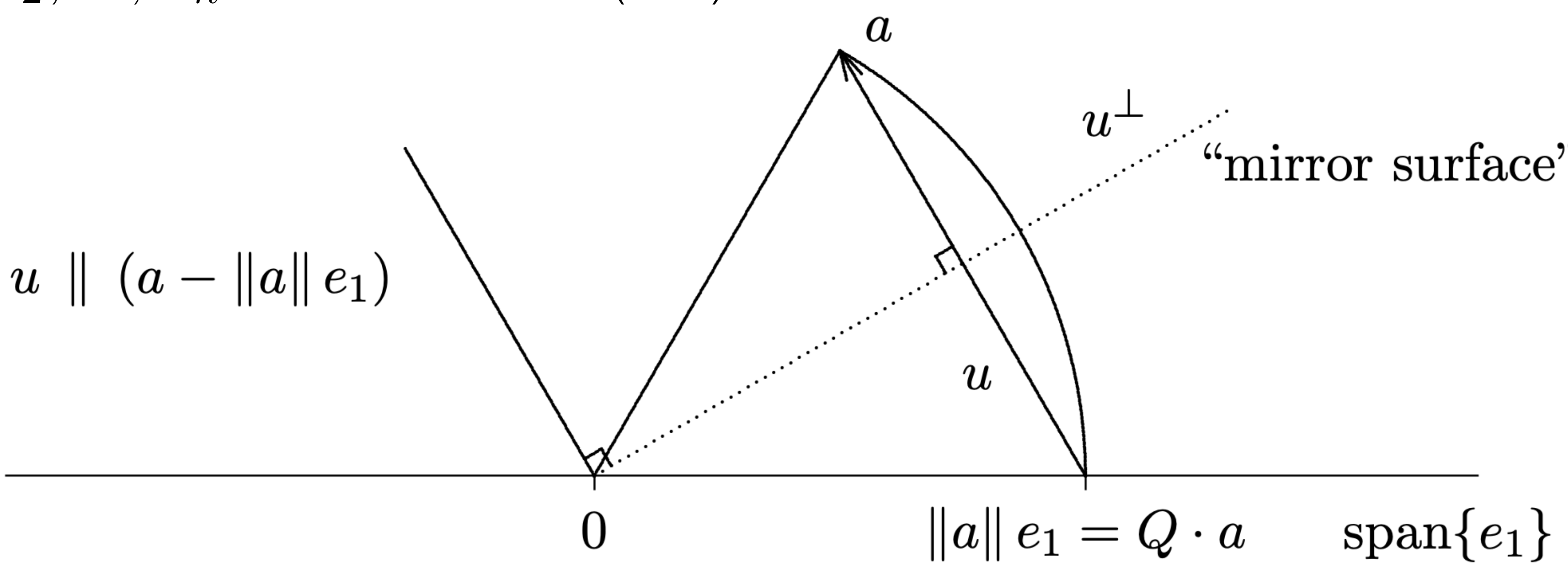
If $w \perp u$ then $Q\,w = (I - 2\,u\,u^T)w = w - 2u\,u^T\,w = w$.

Note that in $n$-space the set of vectors that are orthogonal to one given vector $u \neq o$ form a subspace of dimension $n-1$.

*Students should be able to outline a proof of this statement.*

*Hint :* $\{u^\perp\} = \{w \mid w \perp u\} = \texttt{ker}(u^T)$

Consequently all vectors $w$ that are orthogonal to $u \neq o$ are eigenvectors of $Q = I - 2uu^T$ for the $n-1$ fold eigenvalue $\lambda_2, \ldots, \lambda_n = 1$ since $\dim(u^\perp) = n - 1$.



Here is a 'validation' of this **eigen-based development** of orthogonal elimination matrices $Q$ when implemented on the first column $a$ of a random entry 6 by 6 matrix $A$ in Matlab:

```
>> n = 6; A = randn(n,n), e1 = [1;zeros(n-1,1)]; a = A(:,1);
A =
  -1.1283e+00   4.0112e-01  -1.5409e+00  -3.6378e-01   2.7038e-01  -2.6818e-01
  -1.4245e+00   9.2966e-01  -2.0314e-01  -5.9927e-01  -6.5277e-01  -4.0987e-01
   7.1744e-01  -1.6058e+00  -4.9997e-01  -5.8959e-01   4.7723e-01  -7.1132e-01
  -7.7791e-01   6.6154e-01   3.8302e-01   8.5354e-01  -7.1320e-02   6.1445e-02
   3.1599e-01   2.1385e+00   4.1204e-01  -1.8530e+00  -9.3830e-01  -1.8461e+00
   1.4065e+00   5.4114e-01   4.0549e-01  -2.0730e-01   1.6136e-01  -3.9833e-01
>> na = norm(a), u = a - na*e1; H = eye(n) -2/norm(u)^2*u*u';
na =
   2.5496e+00
>> HA = H*A, Q = qr(A)
HA =
   2.5496e+00  -7.8707e-01   8.1264e-01  -2.7454e-01   3.7383e-01  -3.1978e-01
   4.4409e-16   4.6947e-01   7.0839e-01  -5.6471e-01  -6.1270e-01  -4.2986e-01
  -5.5511e-17  -1.3740e+00  -9.5906e-01  -6.0700e-01   4.5705e-01  -7.0126e-01
   5.5511e-17   4.1022e-01   8.8081e-01   8.7242e-01  -4.9438e-02   5.0532e-02
  -1.1102e-16   2.2406e+00   2.0983e-01  -1.8607e+00  -9.4719e-01  -1.8417e+00
  -2.2204e-16   9.9554e-01  -4.9456e-01  -2.4143e-01   1.2180e-01  -3.7860e-01
Q =
   2.5496e+00  -7.8707e-01   8.1264e-01  -2.7454e-01   3.7383e-01  -3.1978e-01
            0  -2.8789e+00  -6.9105e-01   1.2097e+00   1.0202e+00   1.2925e+00
            0            0   1.4172e+00   1.0693e+00  -3.3158e-01   7.8077e-01
            0            0            0  -1.5369e+00  -5.7189e-01  -1.1922e+00
            0            0            0            0   1.4495e-01  -7.0906e-01
            0            0            0            0            0   1.0018e-01
>> |
```

Note that the first row of $H \cdot A$ and $Q$ agree in the leading 15 digits in Matlab output and that the lower diagonal $n$–1 by $n$–1 block of $Q$ differs greatly from $H \cdot A$'s lower diagonal block when only the first partial QR decomposition was performed.

---

# Applications

## [ 1 ] Orthonormal Bases :

Orthonormal bases of a subspace of $\mathbb{R}^n$ are very important in many applications. The QR factorization of any $n$ by $m$ matrix $A_{n,m} = Q_{n,n} \cdot R_{n,m}$ indicates that the $j$ –th column $a_j$ of $A$ is the matrix times vector product $Q \times$ column $j$ of $R$.
I.e., each column vector $a_j$ of $A$ is a linear combination of the orthonormal columns of $Q$ in positions up to and including $j$.

Therefore the span of the column vectors $a_j$ of $A$ is identical to the span of the column vectors $q_j$ of $Q$.

A minimal orthonormal spanning set or an orthonormal basis of $span\{a_1, ..., a_m\}$ consists of the column unit vectors $q_j$ of $Q$ that have a nonzero pivot in the upper triangular matrix $R$ of the QR factorization $A = QR$.

---

# [ 2 ] Least Squares :

The LR and QR matrix decompositions enable us to solve linear systems $A_{n,m}x = b$. But linear equations can only be solved when $b$ lies in the column space of $A$.

"Least squares" deal with unsolvable linear equations.

If $Ax = b$ cannot be solved because $b$ does not lie in the column space of $A$, least squares methods compute the vector $x$ with $\min_{x \in \mathbb{R}^m} ||Ax - b||$ error.

Here we only deal with $n > m$, i.e. with more equations than unknown in the linear system $A_{n,m}x_m = b_n$ and assume that $A$ has full rank $m$.

This situation occurs naturally when there are more experiments than variables $x_i$, the variables are independent and one wants to find the best solution from long and partially redundant test run data.

If $A_{n,m} = Q_{n,n} * R_{n,m}$ is a QR factorization of $A$ and we want to solve $Ax = b$ as best we can, we start with

$$Q^T Ax = Q^T QAx = Rx = Q^T b$$

and observe that

$$\min_{x\in\mathbb{R}^m} ||Ax - b|| = \min_{x\in\mathbb{R}^m} ||Rx - Q^T b||$$

from Lemma 1.

The upper triangular matrix $R_{n,m}$ contains a nonsingular $m$ by $m$ block $R_1$ on top thanks to our full rank assumption and has only zeros below.

With $R_{n,m} = \begin{pmatrix} R_1 \\ O_{n-m} \end{pmatrix}$ and $Q^T b = \begin{pmatrix} w_m \\ z_{n-m} \end{pmatrix}$ conformally partitioned,

$$\|Ax - b\| = \left\| \begin{pmatrix} R_1 x \\ 0 \end{pmatrix} - \begin{pmatrix} w \\ z_{n-m} \end{pmatrix} \right\|$$

is minimized by the solution $x$ of the nonsingular $m$ by $m$ linear system $R_1 x = w$. And the unavoidable error is $\|z\|$.

When $b$ is not in the column space of $A$ and the system $Ax = b$ cannot be solved, geometry tells us that the perpendicular projection of $b$ onto $\mathrm{im}(A)$ is the closest point $Ax_{LS}$ of $\mathrm{im}(A)$ to $b$.

Let us validate this observation: Is the vector joining $Ax_{LS}$ and $b$ orthogonal to every column of $A$, or is $A^T(Ax_{LS} - b) \approx o_n$?

The Matlab commands below solve one random entry linear system with 40 equations and 5 variables :

```
>> n = 40, m = 5, A=randn(n,m); b=rand(n,1);[Q,R] = qr(A);
R1=R(1:m,1:m); b1=Q'*b; b1m=b1(1:m); xLS= R1\b1m,
errLS=A'*(A*xLS-b),
n =
    40
m =
     5
xLS =
   1.6458e-01
  -7.3331e-02
   1.6630e-01
  -6.3914e-02
  -1.1977e-02
errLS =
  -6.6613e-16
  -8.8818e-16
  -1.5543e-15
   1.3323e-15
   8.8818e-16
>> |
```

The error term for the QR based least squares problem has entries of magnitudes in the $10^{-15}$ or smaller range which indicates proper orthogonality.

In 1822 Carl Friedrich Gauss invented the Normal Equation method for solving least squares problems.

It exploits the orthogonality of the vector from $Ax$ to $b$ to the column space of $A$ directly by multiplying an unsolvable overdetermined linear system $Ax = b$ by $A^T$ from the left and gives us the Normal Equation

$$A^T Ax = A^T b.$$

This form is attractive. It reduces the problem from an $n$ by $m$ system to one with a smaller $m$ by $m$ system matrix.

Recall that typically $n \gg m$ here.

Here is the output of the Gauss' Normal Equation approach for the same 40 by 5 dimensional least squares problem:

```
>> xN=(A'*A)\(A'*b), errN=A'*(A*xN-b),
condA=cond(A), condATA=cond(A'*A)
xN =
   1.6458e-01
  -7.3331e-02
   1.6630e-01
  -6.3914e-02
  -1.1977e-02
errN =
   6.6613e-16
   4.4409e-16
  -1.2212e-15
   6.6613e-16
            0
condA =
   1.5389e+00
condATA =
   2.3683e+00
>>
```

Note that both least squares solution $x_{LS}$ and $x_N$ are nearly identical and the orthogonality of both vectors $Ax_{..} - b$ to $\text{im}(A)$ is similarly maintained.

However, there is a major problem with using Gauss' normal equation. It involves the matrix product of $A^T$ and $A$ and the fact the matrix condition numbers multiply or worsen for matrix products much of the time, due to Olga Taussky (1950).

In our example data set, the condition number of $A$ is relatively benign at around $1.5$. But note that the matrix product $A^T \cdot A$ has a condition number of $2.37 \approx 1.5^2$.

This becomes rather dishabilitating for system matrices $A$ with condition numbers around 100, 1000, or larger. Then the normal equation's solution is likely quite inaccurate and Gauss' theoretically correct normal equation approach is not advised for computations.

# [ 3 ] Deflation Techniques for Krylov Vector Iterations and Eigen Computations :

Here we start from the basis change equation

$$A_{\mathcal{U}} = U^{-1} A_{\mathcal{E}} U \quad (1)$$

and the matrix eigenvalue equation

$$A_{\mathcal{E}} x_{\mathcal{E}} = \lambda x_{\mathcal{E}} \qquad (2)$$

of Lesson Plan 4.

Multiplying equation (1) on the right hand side by $U^{-1}$ we get

$$A_{\mathcal{U}} U^{-1} = U^{-1} A_{\mathcal{E}}$$

and thus

$$A_{\mathcal{U}} U^{-1} x_{\mathcal{E}} = U^{-1} A_{\mathcal{E}} x_{\mathcal{E}} = U^{-1} \lambda x_{\mathcal{E}} = \lambda U^{-1} x_{\mathcal{E}}$$

from equation (2).

Therefore $A_{\mathcal{U}}$ has the same eigenvalue $\lambda$ as $A_{\mathcal{E}}$ and the corresponding eigenvector in $\mathcal{U}$ coordinates is $U^{-1} x_{\mathcal{E}}$.

Next assume that Krylov vector iteration for a square matrix $A_{n,n}$ has found the eigenvector $x$ and eigenvalue $\lambda$ for $A$.

*For Householder matrices $H = I_n - 2uu^T$ with $||u|| = 1$ we have the following Lemma that students should be able to prove.*

**Lemma 4 :** $H = H^{-1} = H^T$.

Given $A_{n,n}$ we assume that normalized Krylov vector iteration for $A$ from a randomly chosen nonzero $n$-vector $b$ has found a unit eigenvector $x$ of $A$.

We want to deflate the eigendata problem for $A_{n,n}$ to one of smaller dimensions $n-1$ by $n-1$.

To do so we will construct a Householder similarity $HAH^{-1}$ on $A$ with $H = I - 2uu^T$ for a still unknown unit vector $u \in \mathbb{R}^n$.

Using Lemma 4 we now block upper triangularize $A$ into the 1 by 1 block $\lambda$ and $A_{2_{n-1,n-1}}$

$$HAH^{-1} = HAH = \begin{pmatrix} \lambda & \cdots & * & \cdots \\ \vdots & & & \\ 0 & & A_2 & \\ \vdots & & & \end{pmatrix}. \qquad (**)$$

Clearly $HAHe_1 = \lambda e_1$ and after multiplying equation $(**)$ on the left by $H$ we obtain

$$AHe_1 = \lambda He_1 \qquad (***)$$

since $H \cdot H = I_n$ by Lemma 4. Thus $He_1$ is an eigenvector of $A$ for the eigenvalue $\lambda$. From Lemma 1, $||He_1|| = ||e_1|| = 1 = ||x||$ and hence $He_1 = \pm\, x$ and the subspaces spanned by $He_1$ and $x$ are identical. Therefore we can ignore the $\pm$ sign.

Formally we need to find a unit vector $u$ and set $H = I_n - 2uu^T$ so that $He_1 = x$ for the Krylov computed unit eigenvector of $A$. By Lemma 5 we know that $H \cdot H = I_n$ and thus $Hx = e_1$. Thus the desired $H$ works exactly as in the elimination method, sending the dense eigenvector $x$ to the first standard unit vector $e_1$.

**Efficient Matrix Coding :**

Matrix splittings and factorizations are great tools to create fast matrix algorithms. Look at equation $(**)$ that shows an explicit upper triangular block decomposition of a matrix $A$. How should we best compute $HAH$ to verify that the eigenvalue $\lambda$ can be found in the $(1,1)$ position of $HAH$ for the properly set up Householder transform $H = I_n - 2uu^T$? How can we compute this double matrix product more efficiently?

The math theoretical way would be to explicitly write

$$HAH = H(AH) = (I_n - 2(u_{n,1} \cdot u_{1,n}^T)) \cdot (A_{n,n} \cdot (I_n - 2(u_{n,1} \cdot u_{1,n}^T))).$$

I.e., form the Householder matrix $H$ and multiply two $n$ by $n$ matrices as indicated at $2\,O(n^3) + O(n^2)$ operations cost. However, when using the splitting of $H = I_n - uu^T$, then $HAH$ can be evaluated at $7\,O(n^2)$ cost:

$$\begin{aligned}
HAH &= HA(I - 2uu^T) = H(A - 2Auu^T) \\
&= (I - 2uu^T)(A - 2Auu^T) \\
&= A - 2Auu^T - 2uu^T(A - 2Auu^T) \\
&= A - 2Auu^T - 2uu^T A + 4uu^T Auu^T = \\
&= A_{n,n} - 2((Au)_{n,1} u_{1,n}^T)_{n,n} - 2(u_{n,1}(u_{1,n}^T A_{n,n})_{1,n})_{n,n} \\
&\qquad + 4(u_{1,n}^T (Au)_{n,1})_{1,1} (u_{n,1} u_{1,n}^T)_{n,n}.
\end{aligned}$$

Note that the 2nd though 4th terms of $HAH$ above require only $2\,O(n^2)$ operations for each column times row evaluation.

# *Epilogue*

Seven Lesson plans, with more subject matter than can be reasonably covered in an introductory one semester Linear Algebra course where **two semesters are really needed !**

This **Modern Matrix Theory** course now ends with some personal and historic memories and observations.

**Episode A:**

*Many decades ago as a graduate student at Caltech I studied with Olga Taussky. Occasionally she directed a Seminar on current results and developments where she gave lectures and the students were to talk on selected topics.*

*One year I was asked to talk about Householder Transforms. Alston had published the first paper on them in 1958 and they were the subject of the very last section 7.8 in his 1964 book on 'The Theory of Matrices in Numerical Analysis'.*

*Working through their algebraic development and equations of the time I felt a void of understanding what Householder Transforms did and how they worked. On Friday, Olga asked how my talk was coming along and I told her of my lack of deep understandings. She told me that Alston would visit Cal-Tech the next week and he might be attending my seminar talk. Over the weekend I restudied and restudied all that I could find out and I conferred with fellow graduate students. We never reached any deep understanding of the why for Householder transformations' success.*

*In my lecture on Tuesday, I started off with their definition and endless formulas.* (see Wikipedia on 'Householder Transformation', e.g.) *Then I left a large part of the blackboard empty and continued - after the blank blackboard gap - with what they did and how they helped in numerical matrix algorithms.*

*That much was quickly done, except for the big gap on the board on which I then mused and asked for help for the rest of the hour. Alston knew of no immediate help, but he acknowledged the issue. Olga smiled benevolently, but my fellow students were at unease - for an embarrassment that they felt.*

*Olga and Alston, however, indicated afterwards that mine was a cherished approach in mathematical research, finding and searching to fill gaps of knowledge.*

*This problem has laid fallow ever since. Until right now, where Householder Transforms have been explained through subspace geometry and matrix eigenanalysis without any need for cumbersome elementary vector algebra or equations.*
*I am happy to have carried this problem with me for decades and now have had the luck to put this 'why' problem to rest.*

*My modernizing educational endeavors have born fruit.*
*I am sure that that this little advance would make Olga and Alston smile today.*

**Episode B:**

On the cost of teaching obsolete math subjects to students just as we were taught and as our teachers were taught and their teachers were taught, .. for centuries, long past.

*Decades ago again, I was up for tenure at Auburn University which was not known for much applied or numerical expertise among its math faculty.*
*I gave a Numerical Linear Algebra course* – a first there – *that fall and sent my students home for the weekend with the homework to try and find all eigenvalues and eigenvectors of the matrix $I_n - 2u\, u^T$ for a unit column vector $u$.*

*On Monday morning I found myself in deep trouble.*

*Unbeknownst to me, some offspring of our faculty had registered for my course and had brought the problem to their parents, aunts and uncles. So they all recalled determinants, the characteristic polynomial, polynomial root finding methods for matrices etc. But none of this was of any use here for the given $n$ by $n$ matrix with a variable $u$ input to boot.*

*I quickly realized that my students should approach this homework differently, but I did not give any hints as to how; I just let it simmer around their dinner tables and dorm corridors ... And when the time came in my course to try orthogonal matrix elimination strategies, the class was much more ready to approach the $I - 2u\,u^T$ eigen problem from Matrix Theory.*

*Faculty dissent and disarray weakened, computational matrix analysis began and I was tenured.*

**Student Episodes C ... :**

Here are the results of the misinformation in 90+ % of our elementary Linear Algebra courses and textbooks:

**(a)** *Engineering students came up to me at the leading Technical University (the RWTH) in Germany: They had to solve a system of linear equations and wanted to do this in Fortran. So they looked at their linear algebra class notes, found* **Cramer's determinantal rule**, *and programmed and tested it for small dimensions $n$. And all went well at first. Their specific problem used $n = 100$ and no output was ever computed. Why? Determinant evaluation is an $O(n!)$ process; Gauss uses $O(n^3)$ operations.* **Why was 'Cramer' taught?**

**(b)** *A leading (top 5) Aerospace Departments (U Illinois Champaign-Urbana) in the US recently asked for a prelim to be retaken by a graduate student because the student had messed up the rule of signs, called '***Ruth-Hurwitz***' (1876, 1895) to determine the stability of a specific 5 by 5 dynamical system.*

*Ruth-Hurwitz requires characteristic polynomial evaluations. But luckily no polynomial root finding. The process goes back to Descartes' Rule of Signs' in La Géométrie (1637).*

*Today one would best load the matrix $A$ into Matlab and use Matlab's* `eig.m` *to plot $A$'s eigenvalues in the complex plane. If all eigenvalues lie in the left half plane, the matrix is stable.*

*This takes less than 50 seconds to implement and milliseconds to run.* **Why do we teach centuries old un-practical stuff?**

*[The prelim was administered and taken on a laptop after all!]*

**(c)** *What if I had not mentioned the limitations of Gauss' normal equation at the end of Lesson Plan 7.*

*What if I had not known or not experienced them myself?*

*How could we avoid such blunders as we subject tenure track faculty increasingly to successfully writing grant applications.*

*Historically it was once an honor to teach beginning math courses. Now nobody can take the time to become up-to-date on the current state of the art in a course's subject matter.*

*This is a structural university wide problem, for math and other areas.*

*I have no answer* *Can you help*