Lesson Plan 5 on Krylov Vector Iteration, Experiments and MatLab Coding

Frank Uhlig

Department of Mathematics and Statistics,
Auburn University, Auburn, AL 36849-5310, USA

uhligfd@auburn.edu

This lesson plan will take 4 class hours to cover basic properties of vector iterations. It proves that all square matrices have eigenvectors and eigenvalues and directs students into simple extensions of the basic Krylov iteration method, possibly on their own.

It is designed to lead students further into computer software and computations and encourages mathematical trial and error investigations and beginning math research.

(version August 3, 2023) iii + 21 p.; 4 - 5 class days

Concepts, Notions and Definitions in Lesson Plan 5

Krylov Subspace

Vector Iteration or Power Method

Matlab Software

Eigenvalues, Eigenvectors

Matlab Coding

Model Equation

Divergence

Normalized Vector Iteration

(continued)

Minimal Polynomial

Fundamental Theorem of Algebra

Roots of Minimal Polynomial

Maximal Modulus Eigenvalue

Shifted Matrix

Other Eigenvalues

Nullspace

Now we begin to include **Matrix Analysis** *in our lessons. The term 'analysis' in math typically deals with computations and estimating errors of algorithms, proving convergence etc. Numerical Analysis depends on computers, software and algorithms rather than on algebraic equation solving that underlies Matrix Theory with its indisputable proofs. This realm gives us theoretically correct equations and processes. Often theoretical formulas can only be realized in theory and they may be of little use for computational math problems.*
We want to experiment with **Vector Iterations** *here to compute eigenvalues of matrices based on Krylov's Vector Iteration ideas and introduce simple Matlab commands and codes. Our 'number crunching approach' in the second half of our elementary Modern Matrix Theory course is very intense. The power of computations will help us in all subsequent* **modern Matrix based Linear Algebra** *lessons.*

Krylov Vector Iteration involves a matrix $A_{n,n}$ and a non-zero vector $b \in \mathbb{R}^n$ or $\mathbb{C}^n$ and uses the set of Krylov vectors

$$\{\underline{b}, \underline{Ab}, \underline{A^2b} \, (= A(Ab)), \underline{A^3b}, ..., \underline{A^{k-1}b}, \underline{A^kb} \, (= A(A^{k-1}b)).$$

Here the exponent k can be any positive integer.

[Krylov subspace based methods today are the main tool for solving linear equations and matrix eigenproblems for high dimensions $n \gg 11,000$ or thereabouts.]

Basic Krylov iterations help us to unlock hidden properties of matrices easily by using software rather than pencil and paper.

Let us start a vector iteration in Matlab right now and observe what happens for a diagonalizable $A_{5,5}$ and b, both with random real entries, for the Krylov vectors as regards A's eigenvalues.

First we define our inputs A and b and determine A's eigenvalues with Matlab's `eig.m` QR algorithm, sorted in decreasing magnitude for comparison.

This is our set-up Matlab code :

```
≫ n = 5, A = randn(n), b = rand(n,1), EA = eig(A),
[ ,m] = sort(abs(EA),'descend'); EA = EA(m), A, b, EA
```

This pair of code lines defines A and b for use in Matlab as

$$A = \begin{pmatrix} 2.5088 & -0.3712 & -0.8951 & 0.3173 & -1.1714 \\ 1.0635 & -0.7578 & -0.4093 & 0.0780 & -1.3853 \\ 1.1569 & -0.5640 & -0.1609 & 1.3244 & 0.3105 \\ 0.0530 & 0.5551 & 0.4093 & -0.2132 & -0.2495 \\ -1.2884 & -0.5568 & -0.9526 & -0.1345 & 0.5037 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0.1182 \\ 0.9884 \\ 0.5400 \\ 0.7069 \\ 0.9995 \end{pmatrix}.$$

The `eig(A)` Matlab command evaluates the five eigenvalues of A as $3.0042, -1.4579, -0.0341+0.8000i, -0.0341-0.8000i, 0.4026 \in \mathbb{C}$ when sorted in decreasing absolute value order.

Here is the Krylov vector set of b, Ab, A^2b, ... for selected powers of A:

b	Ab	A^2b	A^5b	$A^{10}b$	$A^{20}b$	$A^{50}b$	$A^{80}b$
1.1816e-01	-1.500e+00	-2.5520e+00	-7.3153e+01	-1.7355e+04	-1.0400e+09	-2.2325e+23	-4.7923e+37
9.8842e-01	-2.174e+00	8.9869e-01	-4.3445e+01	-9.2270e+03	-5.5456e+08	-1.1904e+23	-2.5553e+37
5.3998e-01	7.3896e-01	-3.8181e-01	-2.1290e+01	-4.8777e+03	-2.9277e+08	-6.2846e+22	-1.3491e+37
7.0692e-01	3.7595e-01	-8.6218e-01	-1.2245e+01	-3.5345e+03	-2.1072e+08	-4.5233e+22	-9.7097e+36
9.9949e-01	-8.0854e-01	1.9814e+00	4.9581e+01	1.3088e+04	7.8225e+08	1.6792e+23	3.6045e+37

What is happening? Why can Krylov vectors get so huge?

Time for student discussions among themselves, their ideas, explanations ..., observations, ...

Back to arithmetic here : Heuristically, the ten power iterations from $A^{10}b$ to $A^{20}b$ (for example) have increased the resulting iterates by a factor of around 10^5 in magnitude from around 10^4 to 10^9. How big is the average increase x per iteration?

$$x^{10} \approx 10^5$$

is the model equation for the average Krylov vector growth.

Taking logarithms on both sides, we have $10 \cdot \log_{10}(x) \approx 5$ or $\log_{10}(x) = 1/2$ and $x \approx 10^{1/2} \approx 3$ since $3^2 = 9$.

What about the magnitude increases in the 30 step transitions from $A^{20}b$ to $A^{50}b$ or from $A^{50}b$ to $A^{80}b$ in our data table? Both Krylov iterates increase by around 10^{14} and they lead to the arithmetic model equation $y^{30} \approx 10^{14}$.

Thus $30 \cdot \log_{10}(y) \approx 14$, $\log_{10}(y) \approx 1/2$ and $y \approx 3$ again – with some grains of salt.

Something mathematical is clearly happening here!

Switching to matrix analysis mode, vector iteration is clearly diverging here. Why ? How can matrix algebra help?

Our given matrix $A_{5,5}$ has five distinct eigenvalues, sorted by magnitude $|\lambda_1|(\approx 3) > |\lambda_2| > ... > |\lambda_5|$ as Matlab has shown.

Therefore A is diagonalizable for its complex eigenvector basis $\mathcal{U}$ of $\mathbb{C}^5$, collected column-wise in $U_{5,5}$.

The theory based eigen-equation for A is $AU = UD$ with $D = diag(\lambda_j)$ diagonal for A's eigenvalues λ_j.

According to Matrix algebra each vector $b \neq o_n$ is a unique and non-zero linear combination $b = \sum_{j=1}^{n} \alpha_j u_j$ of the eigenvector basis $\mathcal{U} = \{u_1, ..., u_n\}$ for A.

In the following simplistic algebraic deduction we assume that $\alpha_1 \neq 0$:

$$\begin{aligned} A^\ell b &= A^\ell \sum_{j=1}^{n} \alpha_j u_j = \sum_{j=1}^{n} \alpha_j A^\ell u_j \\ &= \sum_{j=1}^{n} \alpha_j \lambda_j^\ell u_j \qquad (*) \\ &= \alpha_1 \lambda_1^\ell \left(u_1 + \frac{\alpha_2}{\alpha_1} \left(\frac{\lambda_2}{\lambda_1} \right)^\ell u_2 + \cdots + \frac{\alpha_n}{\alpha_1} \left(\frac{\lambda_n}{\lambda_1} \right)^\ell u_n \right). \end{aligned}$$

In formula $(*)$, the fractions $\left|\frac{\lambda_j}{\lambda_1}\right| < 1$ for all $j = 2, ..., n$ and thus – analytically speaking again – the powers $A^\ell b$ converge to $\alpha_1 \lambda_1^\ell u_1$.

If $|\lambda_1| > 1$ – as is the case for our $A_{5,5}$ and $\alpha_1 \neq 0$ in b then the Krylov vector iterations $A^\ell b$ clearly diverge to infinity or to an infinitely large multiple of the first basis vector u_1 for A.

Matrix Algebra now will rescue us from this growth dilemma: If we were to normalize the ever increasing vectors $A^\ell b$ when $\lambda_1 > 1$ and replace it by

$$\frac{A^\ell b}{\|A^\ell b\|}$$

at every power ℓ, then the iterates $A^\ell b / \|A^\ell b\|$ would always be unit vectors and convergence could be easily read off the data.

b	$Ab/\|\|Ab\|\|$	$A^2b/\|\|A^2b\|\|$	$A^{10}b/\|\|A^{10}b\|\|$	$A^{20}b/\|\|A^{20}b\|\|$	$A^{50}b/\|\|A^{50}b\|\|$	$A^{80}b/\|\|A^{80}b\|\|$
1.1816e-01	-7.3257e-01	-7.3146e-01	-7.1254e-01	-7.1241e-01	-7.1241e-01	-7.1241e-01
9.8842e-01	2.5798e-01	-4.9238e-01	-3.8048e-01	-3.7987e-01	-3.7987e-01	-3.7987e-01
5.3998e-01	-1.0960e-01	-3.1935e-01	-2.0075e-01	-2.0055e-01	-2.0055e-01	-2.0055e-01
7.0692e-01	-2.4749e-01	-8.3940e-03	-1.4401e-01	-1.4434e-01	-1.4434e-01	-1.4434e-01
9.9949e-01	5.6879e-01	3.4709e-01	5.3524e-01	5.3584e-01	5.3584e-01	5.3584e-01

Thus normalized Krylov vector iterations converge quickly to the dominant eigenvector u_1 of A that was computed for the max modulus eigenvalue λ_1 of A by Matlab earlier.

How can we find the eigenvalue λ_1 of A from the normalized eigenvector of $A^{80}b/||A^{80}b||$ for example?

By definition $Au_1 = \lambda_1 u_1$. In double precision this means for

$$u_1 = \begin{pmatrix} -7.124070069031886e-01 \\ -3.798708481611794e-01 \\ -2.005483353698798e-01 \\ -1.443426438954290e-01 \\ 5.358357598800159e-01 \end{pmatrix}$$ and λ_1 the following :

$$Au_1 = A \begin{pmatrix} -7.124070069031886e-01 \\ -3.798708481611794e-01 \\ -2.005483353698798e-01 \\ -1.443426438954290e-01 \\ 5.358357598800159e-01 \end{pmatrix} = \lambda_1 \begin{pmatrix} -7.124070069031886e-01 \\ -3.798708481611794e-01 \\ -2.005483353698798e-01 \\ -1.443426438954290e-01 \\ 5.358357598800159e-01 \end{pmatrix} = \lambda_1 u_1,$$

or expressed in Matlab notation for each component of u_1 :

$$\lambda_1 = Au_1./u_1 = \begin{pmatrix} 3.004173549878356 \\ 3.004173549878356 \\ 3.004173549878357 \\ 3.004173549878356 \\ 3.004173549878356 \end{pmatrix}$$ in each component, which agrees almost verbatim with Matlab's computed leading eigenvalue 3.004173549878358 for A.

How do we know that all matrices $A_{n,n}$ have eigenvalues?

Look at the Krylov vector progression b, Ab, A^2b, ..., A^nb for $b \neq o_n$. These are $n+1$ vectors in $\mathbb{R}^n$ oder $\mathbb{C}^n$ and thus they are linearly dependent with maximally n pivots in their associated n by $n+1$ column vector matrix's row echelon form. I.e.,

$$A^k b = c_0 b + c_1 Ab + \cdots + c_{k-1}A^{k-1}b$$

for some $k \leq n$ where not all c_i are 0. Thus

$$\left(A^k - c_{k-1}A^{k-1} - \cdots - c_1 A - c_0 I_n\right) b = o_n \ . \qquad (**)$$

Now we study the polynomial

$$p_A(x) = x^k - c_{k-1}x^{k-1} - \cdots - c_1 x - c_0.$$

$p_A(x)$ is called the minimal polynomial of A if k is minimal for A where $k \leq n$ the first column in the Krylov progression matrix's row echelon form without a pivot.

Next we use the Fundamental Theorem of Algebra.

Fundamental Theorem of Algebra
Every real or complex polynomial $p(x) = \sum_{j=k}^{j=0} a_j x^j$ *of degree* k *has* k *roots* $x_i \in \mathbb{C}$ *with* $p(x_i) = 0$.
I.e., $p(x) = a_k(x - x_k) \cdots (x - x_1)$ *with* k *possibly repeated roots* x_j.

Thus
$$\begin{aligned} p_A(A) &= A^k - c_{k-1}A^{k-1} - \cdots - c_1 A - c_0 I_n \\ &= (A - x_k I_n) \cdots (A - x_1 I_n) \end{aligned}$$

for the roots x_i of A's minimal polynomial $p_A(x)$.
From equation $(**)$ and

$$p_A(A)b = (A - x_k I_n) \cdots (A - x_1 I_n)b = o_n \text{ for } b \neq 0 \quad (***)$$

the matrix product $p_A(A) = (A - x_k I_n) \cdots (A - x_1 I_n)$ is singular.

What about the individual factors $A - x_i I_n$?

Can any one of these be nonsingular?

Clearly any two degree one factors of $p_A(A)$ commute since

$$(A - x_\ell I_n)(A - x_m I_n) = (A - x_m I_n)(A - x_\ell I_n)$$

for all ℓ and m by inspection.

If one factor matrix $A - x_\ell I_n$ is nonsingular then it can be moved up to the front of writing $p_A(A)$ and

$$q(x) = p_A(x)/(x - x_\ell)$$

has degree $k - 1$. Then

$$q(A)\, b = (A - x_\ell I_n)^{-1} p_A(A)\, b \neq o_n$$

since $(A - x_\ell I_n)^{-1}$ is nonsingular as well and $b \neq o_n$.

This makes q a minimal polynomial for A of lower degree than k which contradicts the minimal property of p_A.

Thus each factor $A - x_i I_n$ of $p_A(A)$ is singular and A has at least k eigenvectors for the k eigenvalues x_i.
In other words, A has at least one eigenvector in the nullspace of $A - x_i I_n$ for each $i = 1, ..., k \leq n$.

We have spent a considerable amount of time and space to shore up our theoretical knowledge of eigenproperties of matrices.

Unfortunately the minimal polynomial of a matrix $A_{n,n}$ cannot help us in any way to compute those pesky – elusive for 150 years – matrix eigenvectors and eigenvalues.

Krylov's vector iteration method allows us to compute additional eigenvalues besides the max modulus one.

If λ_1 is the maximal modulus eigenvalue of $A_{n,n}$ and u_1 the associated eigenvector, then $B = A - \lambda_1 I_n$ has the eigenvalue 0 for the same eigenvector u_1.

This statement is a worthy exercise for all students to work through theoretically and try to prove that this is true.

A Krylov iteration with $B = A - \lambda_1 I_n$ and the same vector $b \neq o_n$ computes an eigenvector v_B for B and B's maximal modulus eigenvalue μ_B.

Then $\lambda_2 = \lambda_1 + \mu_B$ is an eigenvalue of A with v_B as associated eigenvector.

Students, please check this !

Here are the numerical results of Krylov normalized vector iterations for our original example matrix $A_{5,5}$, b, and B.

$B = A - 3.006... * I_5$ has the nonzero eigenvalue $8.85... \cdot 10^{-16}$ near zero under Matlab which is comforting and expected.
The dominant eigenvalue $4.46211...$ of $B = A - \lambda_1 I_n$ under Krylov vector iteration with b is $\mu_B = 4.46211....$

This translates to a second eigendata pair of A at the eigenvalue $\lambda_2 = \lambda_1 + \mu_B = -1.457941230309017$ for the vector iteration computed eigenvector v_B. This agrees in all but the two last digits with the eigen-data computed for A via Matlab.

For diagonalizable real 3 by 3 matrices $A_{3,3}$ we have also been able to compute the third eigenvalue and its eigenvector via Krylov vector iteration.

For example for the random entry matrix $A_{3,3} =$

$$\begin{pmatrix} -1.203268186415017 & -0.1729138442591677 & -3.232037795940007 \\ 1.037815639485240 & -1.208652054740199e & -1.086959229461330 \\ -0.8459442123361178 & -0.2.971267999954044 & -1.426436159479723 \end{pmatrix},$$

Matlab computes A's three eigenvalues as

$$0.5160221986685007, -2.867153146123481, -1.487225453179958.$$

The first two eigenvalues λ_1 and λ_2 above and their associated eigenvectors u_1 and u_2 are replicated exactly by using Krylov vector iteration for A and $B = A - \lambda_1 I_3$ from any starting vector $b \neq o_3$ as we have just learnt to do, except for the very last digit.

To locate the third eigenvalue of $A_{3,3}$, we find a column vector w_3 in $\mathbb{R}^3$ that lies in the nullspace of $V = \begin{pmatrix} \cdots & u_1^T & \cdots \\ \cdots & u_2^T & \cdots \end{pmatrix}_{2,3}$.

Then we operate with $W = \begin{pmatrix} \vdots & \vdots & \vdots \\ u_1 & u_2 & w \\ \vdots & \vdots & \vdots \end{pmatrix}_{3,3}$ on our 3 by 3 example matrix A and form the matrix product $UT = W^{-1} * A * W$ it becomes $UT =$

$$\begin{pmatrix} -2.867153146123482 & -3.46944 \cdot 10^{-16} & -1.521078755080422 \\ 1.11022 \cdot 10^{-16} & 0.5160221986685007 & -1.852060317168440 \\ 1.11022 \cdot 10^{-16} & 2.49800 \cdot 10^{-16} & -1.487225453179958 \end{pmatrix}.$$

Miraculously UT is upper triangular (therefore its name UT) when we set the minuscule entries of magnitudes around 10^{-16} in UT equal to zero.

All three eigenvalues of A sit on the diagonal of

$$UT_{\#} = \begin{pmatrix} -2.867153146123482 & 0 & -1.521078755080422 \\ 0 & 0.5160221986685007 & -1.852060317168440 \\ 0 & 0 & -1.487225453179958 \end{pmatrix}.$$

We already know the eigenvectors u_1 and u_2 for $\lambda_1 \approx -2.867$ and $\lambda_2 \approx 0.516$ of A.

How to find an eigenvector for the third eigenvalue $\lambda_3 \approx -1.4877$ of A?

Note : For every eigenvalue λ of a square matrix $A_{n,n}$, the matrix $A - \lambda I_n$ is singular and each vector in its kernel or nullspace is an eigenvector of A.

Every student should verify the last sentence now.

Thus we only need to find the vectors in the nullspace of $A - \lambda I_n$. We could do this via a row reduction and solve the homogeneous linear system system $(A - \lambda I_n)x = o_n$. Or we could use Matlab which has a built-in nullspace function `null.m`.

Doing the latter and using Matlab on the Krylov vector iteration data evaluates the eigenvector w for the eigenvalue $\lambda_3 \approx -1.4877$ of our example matrix $A_{3,3}$ as

$$w = \begin{pmatrix} 0.3354286828360653 \\ -0.9386890693559886 \\ 0.0796895840273395 1 \end{pmatrix},$$

exactly as Matlab's `eig.m` computed w via Francis' QR algorithm.

At the very end of Lesson Plan 7, we will pick up our recent thoughts and efforts to find additional matrix eigen-information from Krylov.
There we will learn how to compute multiple matrix eigenvectors and eigenvalues by using Krylov iterations combined with Householder transform induced basis changes. This method will let us deflate the n by n matrix eigen problem, one dimension at a time.

Lesson Plan 6 studies angles between vectors in $\mathbb{C}^n$*, orthogonal vectors and orthonormal bases, orthogonal and unitary matrices* Q*, as well as special orthogonal elimination matrices such as Householder transformations* H*. Householder transforms play a key role in computing orthogonal bases of subsets, in factorizing general square matrices* $A_{n,n} \in \mathbb{C}_{n,n}$ *as* $Q \cdot R$ *with* R *upper triangular, and in solving the complete matrix eigenproblem via Krylov vector iteration.*