ME 500
Numerical Methods in Mechanical Engineering
Assignment 4

Brandon Lampe

November 3, 2015

1 Written summary of material and definitions

eigenproblem When a matrix or second-order tensor (1) operates on a vector 2 and the result is a scalar times the same vector, then that vector is an eigenvector and the scalar is an eigenvalue. The standard eigenproblem is:

given a matrix, find $\{p\}$ *and* λ *that satisfies* $\big[[A] - \lambda[I]\big]\{p\} = 0$ (or Equations 3 and 4 in direct notation)

The general eignproblem is defined:
given the matrices [A] and [B], find $\{p\}$ *and* λ *that satisfies* $\big[[A] - \lambda[B]\big]\{p\} = 0$

The characteristic polynomial $p(\lambda)$ is written as Equation 6, and the characteristic equation results when the characterisic polynomial is set equal to zero (Equation 7), Where the roots of the characteristic equation are the eigenvalues. The Cayley-Hamilton Theorem states that a tensor satisfies it's own characteristic equation; i.e., the roots of the characteristic polynomial can be obtained by using the three sets of independent invariants associated with a 3×3 tensor.

Components of the eigenvector associated with an eigenvalue may be obtained obtained by substiting back in the eigenvalue (λ) into Equation 5 and solving for the components of the eigenvector $\{p\}$. Components of an eigenvector are not independent; therefore, one of the components may be arbitrarily chosen (say $p_{1,1} = 1$) and the remaining components may be solve for. Also, for real-symmetric matrices each eigenvalue is typically associated with a unique eigenvector, that is λ_1 solves for $\overset{e}{p}_{1,1}, \overset{e}{p}_{1,2}, \overset{e}{p}_{1,3}$.

For a real-symmetric matrix, each eigenvector is orthogonal to all other eigenvectors; therefore, for a 3×3 matrix, if two eigenvectors are obtained, the third eigenvector must be orthogonal to the first two and may be solved for via a cross product. When a tensor is expressed in the basis composed of its eigenvectors, this tensor is said to be in its principal basis and all off-diagonal components of the tensor will equal zero. The modal matrix is a matrix which has columns composed of the eigenvectors, this matrix also acts as a transformation matrix between the "lab" (e-e) basis and principal basis (p-p) as shown in Equation 10.

An eigenpair consists of an eigenvalue and its associated eigenvector. The eigensystem of a matrix is the complete set of all eigenpairs. The modal matrix, composed of a set of normalized eigenvectors, is an orthonormal matrix. A matrix with all positive eigenvalues is positive definite and its inverse exists.

Spectral decomposition of a matrix is expressed in Equation 11 shown using the outer product of eigenvectors (repeated indices are not summed) and modal matrices and diagonal matrix of eigenvalues, respectively. The function of matrix, is defined as the specific function acting on the spectrally decomposed matrix, e.g., Equation 12 shows the the square root of $[A]$ is obtained.

$$[A] \implies A_{ij}\boldsymbol{e}_i \otimes \boldsymbol{e}_j \tag{1}$$

$$\{p\} \implies p_k\boldsymbol{e}_k \tag{2}$$

$$\boldsymbol{A} \cdot \boldsymbol{p} = \lambda\boldsymbol{p} \tag{3}$$
$$[\boldsymbol{A} - \lambda\boldsymbol{I}] \cdot \boldsymbol{p} = \boldsymbol{0} \tag{4}$$

$$\left[\overset{e-e}{[A]} - \lambda[I]\right]\overset{e}{\{p\}} = \{0\} \tag{5}$$

$$det\left(\left[\overset{e-e}{[A]} - \lambda[I]\right]\right) = p(\lambda) \tag{6}$$

$$p(\lambda) = 0 \tag{7}$$

$$\left[\overset{e-e}{[A]} - \lambda_1[I]\right]\begin{Bmatrix} \overset{e}{p}_{1,1} \\ \overset{e}{p}_{1,2} \\ \overset{e}{p}_{1,3} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{8}$$

$$\boldsymbol{p}_1 = \overset{e}{p}_{1,1}\boldsymbol{e}_1 + \overset{e}{p}_{1,2}\boldsymbol{e}_2 + \overset{e}{p}_{1,3}\boldsymbol{e}_3 \tag{9}$$

$$\overset{e-p}{[M^o]} = \begin{bmatrix} \overset{e}{p}_{1,1} & \overset{e}{p}_{2,1} & \overset{e}{p}_{3,1} \\ \overset{e}{p}_{1,2} & \overset{e}{p}_{2,2} & \overset{e}{p}_{3,2} \\ \overset{e}{p}_{1,3} & \overset{e}{p}_{2,3} & \overset{e}{p}_{3,3} \end{bmatrix} \tag{10}$$

$$[A] = \sum_{i=1}^{n} \lambda_i \{p_i\} < p_i >= [M^o]^T[\lambda][M^o] \tag{11}$$

$$[A]^{1/2} = \sum_{i=1}^{n} \lambda_i^{1/2} \{p_i\} < p_i > \tag{12}$$

rank r, is the number of independent vectors in a given vector space, which may be obtained via the Gram-Schmidt procedure.

$$\text{for } [A]_{m\times n} \ r \leq n$$

nullspace $N([A])$, a vector space of $[A]$ formed from the solution of $[A]\{x\} = 0$. Where the *nullspace* of [A] consists of all vectors $\{x\}$ which satisfy this equation

range $R([A])$, a vector space formed by the columns of a matrix, also known as the column space, which is a subspace of R^m (the whole vector space). This space includes all vectors not included in the nullspace.

Rayleigh's quotient defined as:

$$R(\{x\}) = \frac{< x > [A]\{x\}}{< x > \{x\}}$$

where, if $\{x\}$ is an eigenvector, then $R(\{x\}^i) = \lambda_i$

Gershgorin's theorem Every eigenvalue of [A] lies in at least one of the circles C_i, which are centered at the diagonal entries of [A] (e.g. A_{ii}) and have have radii of

$$r_i = \sum_{i\neq j} |a_{ij}|$$

condition number when associated with the linear algebraic problem $[A]\{x\} = \{b\}$, the condition number provides a bound on how inaccurate the solution $\{x\}$ will be when it is numerically approximated. This is a property of the matrix [A], not the not the chosen algorithm. The condition number (C) may be calculated as the ratio between the maximum and mininimum eigenvalues:

$$C = \frac{\lambda_{max}}{\lambda_{min}}$$

2

Hand calculations are attached at the end of this document.

2) for $[A]$ find:

a) eigenpairs

b) Modal matrix, c) calc. $[M^o]^T[A][M^o]$, $[M^o][M^o]^T$, $\sum \lambda_i \underline{e}^i \underline{e}^i$

d) rank and range of null space

i) $[A] = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$

eigenproblem : $[[A] - \lambda[I]]\{e\} = \{0\}$

Cayby-Hamilton: $\det([A] - \lambda[I]) = 0$

$$\det\left(\begin{bmatrix} 1-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{bmatrix}\right) = (1-\lambda)(2-\lambda)(1-\lambda) - (1-\lambda) - (1-\lambda)$$

$$= (1-2\lambda+\lambda^2)(2-\lambda) - 2 + 2\lambda$$

$$= \cancel{2} - 4\lambda + \cancel{2\lambda^2} - \lambda + \cancel{2\lambda^2} - \cancel{\lambda^3} - \cancel{2} + 2\lambda$$

$$= -\lambda^3 + 4\lambda^2 - 3\lambda$$

$$= \lambda(-\lambda^2 + 4\lambda - 3)$$

- characteristic eqn $\rightarrow$ $\lambda(-\lambda^2+4\lambda-3) = 0$
- solve for the roots:

$\lambda_1 = 0$

$$\frac{-4 \pm \sqrt{16 - 12}}{-2} \rightarrow \frac{-4 \pm 2}{-2} = \lambda_2, \lambda_3$$

$\lambda_2 = 1$, $\lambda_3 = 3$

- solve for the eigenvectors $\Rightarrow$ sub. λ back into eigenproblem:

for λ_1; solve for $\underline{e}^1$

$$\begin{bmatrix} 1-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{bmatrix} \begin{Bmatrix} e^1_1 \\ e^1_2 \\ e^1_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}, \quad \text{Let } \underline{e}^1_1 = 1 \text{ and } \lambda = 0$$

$\underline{e}^1_2 = 1$, $\underline{e}^1_3 = 1$

cont. $\underline{e}^1$:

normalize $\underline{e}^1$: $\dfrac{\{\underline{e}^1\}}{\|\{\underline{e}^1\}\|_2} = \dfrac{\langle 1, 1, 1\rangle}{\sqrt{1^2+1^2+1^2}}$

$= \left\langle \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right\rangle$

solve for $\underline{e}^2$:

$$\begin{bmatrix} 1-\lambda_2 & -1 & 0 \\ -1 & 2-\lambda_2 & -1 \\ 0 & -1 & 1-\lambda_2 \end{bmatrix} \begin{Bmatrix} e^2_1 \\ e^2_2 \\ e^2_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}, \quad \lambda_2 = 1$$

$$\left.\begin{aligned} &\text{Let } e^2_1 = 1 \\ &-e^2_2 = 0 \\ &-1 + 0 - 1e^2_3 = 0 \\ &\qquad e^2_3 = -1 \end{aligned}\right\} \text{normalize: } \frac{\underline{e}^2}{\sqrt{1^2+0^2+1^2}}$$

$\underline{e}^2 = \left\langle \frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}} \right\rangle$

- solve for $\underline{e}^3$:

$$\begin{bmatrix} 1-\lambda_3 & -1 & 0 \\ -1 & 2-\lambda_3 & -1 \\ 0 & -1 & 1-\lambda_3 \end{bmatrix} \begin{Bmatrix} e^3_1 \\ e^3_2 \\ e^3_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \qquad \begin{aligned} &\lambda_3 = 3 \\ &\text{Let } e^3_1 = 1 \end{aligned}$$

$-2 - 1e^3_2 = 0 \quad \rightarrow \quad e^3_2 = -2$

$2 - 2e^3_3 = 0 \quad \rightarrow \quad e^3_3 = 1$

normalize: $\dfrac{\underline{e}^3}{\sqrt{1^2+0^2+1^2}} \Rightarrow \underline{e}^3 = \left\langle \frac{1}{\sqrt{6}}, \frac{-2}{\sqrt{6}}, \frac{1}{\sqrt{6}} \right\rangle$

$\therefore$ a) the eigenpairs are $(\lambda_i, \underline{e}^i)$:

$0, \left\langle \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right\rangle$

$1, \left\langle \frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}} \right\rangle$

$3, \left\langle \frac{1}{\sqrt{6}}, \frac{-2}{\sqrt{6}}, \frac{1}{\sqrt{6}} \right\rangle$

b) the modal matrix:

$$\overset{a-p}{[M^0]} = \left[\frac{1}{\sqrt{3}}\begin{Bmatrix}1\\1\\1\end{Bmatrix} \quad \frac{1}{\sqrt{2}}\begin{Bmatrix}1\\0\\-1\end{Bmatrix} \quad \frac{1}{\sqrt{6}}\begin{Bmatrix}1\\-2\\1\end{Bmatrix} \right]$$

c)

$$\overset{p-a}{[M^0]}^T \overset{a-p}{[M^0]} = \begin{bmatrix} \frac{1}{\sqrt{3}} \langle 1 & 1 & 1 \rangle \\ 1/\sqrt{2} \langle 1 & 0 & -1 \rangle \\ 1/\sqrt{6} \langle 1 & -2 & 1 \rangle \end{bmatrix} \left[\frac{1}{\sqrt{3}}\begin{Bmatrix}1\\1\\1\end{Bmatrix} \frac{1}{\sqrt{2}}\begin{Bmatrix}1\\0\\-1\end{Bmatrix} \frac{1}{\sqrt{6}}\begin{Bmatrix}1\\-2\\1\end{Bmatrix} \right]$$

$$= \begin{bmatrix} (1/3 + 1/3 + 1/3) & 0 & 0 \\ 0 & (\frac{1}{2} + 0 + \frac{1}{2}) & 0 \\ 0 & 0 & (\frac{1}{6} + \frac{4}{6} + \frac{1}{6}) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Bigg\} \text{ because eigenvectors are orthonormal} \Rightarrow [M^0]^T = [M^0]^{-T}$$

$$\overset{p-a}{[M^0]}^T \overset{a-a}{[A]} \overset{a-p}{[M^0]} = \overset{p-p}{[A]} \Rightarrow \overset{p-p}{[A]} \text{ is in the principal basis}$$

$$[M^0]^T[A] = \begin{bmatrix} (\frac{1}{\sqrt{3}} + -\frac{1}{\sqrt{3}}) & (-\frac{1}{\sqrt{3}} + 2/\sqrt{3} - 1/\sqrt{3}) & (0 + 1/\sqrt{3} - 1/\sqrt{3}) \\ (+1/\sqrt{2} + 0 + 0) & (-1/\sqrt{2} + 0 + 1/\sqrt{2}) & (0 + 0 - 1/\sqrt{2}) \\ (1/\sqrt{6} + 2/\sqrt{6} + 0) & (-1/\sqrt{6} - 4/\sqrt{6} - 1/\sqrt{6}) & (0 + 2/\sqrt{6} + \frac{1}{\sqrt{6}}) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ +1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 3/\sqrt{6} & -6/\sqrt{6} & 3/\sqrt{6} \end{bmatrix}$$

$$[M^0]^T[A][M^0] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2} + \frac{1}{2} & 0 \\ 0 & 0 & (\frac{3}{6} + \frac{12}{6} + \frac{3}{6}) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

$$= \lambda [I]$$

$$\sum_{i=1}^{n} \lambda_i \{e^i\} \langle e^i \rangle = [0] + 1 \begin{bmatrix} 1/2 & 0 & -1/2 \\ 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \end{bmatrix} + 3 \begin{bmatrix} 1/6 & -2/6 & +1/6 \\ -2/6 & 4/6 & -2/6 \\ 1/6 & -2/6 & 1/6 \end{bmatrix}$$

$\therefore \sum_{i=1}^{\wedge} \lambda_i \{e^i\}\langle e^i\rangle = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} = [A]$

d) rank => number of nonzero eigenvalues

$\therefore r = 2$

=> only 2 of the 3 rows are linearly independent

i.e. $-1 \cdot \langle A^1\rangle + -1 \cdot \langle A^3\rangle = \langle A^2\rangle$

range => the column space of [A]

this is the plane $\underline{e}^2$ + $\underline{e}^3$ lie in and $\underline{e}^1$ is orthogonal to $\therefore$ it may be described by a vector $\perp$ to it:

range = $\underline{e}^2 \times \underline{e}^3$

nullspace of [A] => the vector space of [A] that is not indepentent

$\therefore$ $\underline{e}^1$ is the nullspace of [A]

2. b)

$$[A] = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

eigenproblem: $\Big[[A] - \lambda [I]\Big]\{e\} = \{0\}$

-> because the (2,2) component of [A] is the only nonzero component on the row, 2 is an eigenvalue

$\lambda_1 = 2$

$$\rightarrow \begin{bmatrix} 1-\lambda & 0 & -1 \\ 0 & 2-\lambda & 0 \\ -1 & 0 & 1-\lambda \end{bmatrix} \Big\{ e \Big\} = \Big\{ 0 \Big\}$$

$$\det\Big(\big[[A]-\lambda[I]\big]\Big) = (1-\lambda)(2-\lambda)(1-\lambda) - (2-\lambda) \Rightarrow \text{Char. Poly.}$$

$$\underbrace{(2-\lambda)}_{\lambda_1 = 2}\underbrace{\Big[(1-\lambda)(1-\lambda) - 1\Big]}_{\text{solve for roots}} = 0 \Rightarrow \text{char. Eqn.}$$

$$1 - 2\lambda + \lambda^2 - 1 = 0$$

$$\lambda^2 - 2\lambda = 0$$

$$\frac{2 \pm \sqrt{4-0}}{2}$$

$$\frac{2 \pm 2}{2} = \lambda_2, \lambda_3 \rightarrow \begin{matrix} \lambda_2 = 2 \\ \lambda_3 = 0 \end{matrix}$$

-> find eigenvectors: solve for $\underline{e}^1$ ->

by observation, $\underline{e}^1 = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$ will satisfy the char. Eqn.

solve for $\underline{e}^2$ ->

$$\begin{bmatrix} 1-\lambda & 0 & -1 \\ 0 & 2-\lambda & 0 \\ -1 & 0 & 1-\lambda \end{bmatrix} \begin{Bmatrix} e_1^2 \\ e_2^2 \\ e_3^2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \quad \text{for } \lambda_2 = 2, \text{ Let } e_1^2 = 1$$

$$-1 - 1e_3^2 = 0 \rightarrow e_3^2 = -1$$

$$0e_2^2 = 0 \rightarrow e_2^2 = 0 \quad \therefore \underline{e}^2 = \begin{Bmatrix} 1 \\ 0 \\ -1 \end{Bmatrix} = \begin{Bmatrix} 1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{Bmatrix}$$

solve for $\underline{e}^3 \rightarrow$

$$\begin{bmatrix} 1-\lambda & 0 & -1 \\ 0 & 2-\lambda & 0 \\ -1 & 0 & 1-\lambda \end{bmatrix} \begin{Bmatrix} e^3_1 \\ e^3_2 \\ e^3_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \qquad \text{for } \lambda_3 = 0, \text{ Let } e^3_1 = 1$$

$$1 - 1e^3_3 = 0 \quad \rightarrow \quad e^3_3 = 1$$

$$2e^3_2 = 0 \quad \rightarrow \quad e^3_2 = 0 \qquad \therefore \quad \underline{e}^3 = \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{Bmatrix}$$

a) the eigenpairs are then:

$2, \ \langle 0, 1, 0 \rangle$

$2, \ \langle 1/\sqrt{2}, 0, -1/\sqrt{2} \rangle$

$0, \ \langle 1/\sqrt{2}, 0, 1/\sqrt{2} \rangle$

b) the modal matrix:

$$\overset{a-p}{[M^0]} = \left[\begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \quad 1/\sqrt{2} \begin{Bmatrix} 1 \\ 0 \\ -1 \end{Bmatrix} \quad 1/\sqrt{2} \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix} \right]$$

c) $$\overset{p-a}{[M^0]}^T \overset{a-a}{[A]} \overset{a-p}{[M^0]} = \overset{p-p}{[A]} \quad \Rightarrow \quad \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$, in the principal basis

$$[M^0]^T [M^0] = [I] \quad \Rightarrow \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\sum_{i=1}^{n} \lambda_i \{e^i\}\{e^i\}^T = 2\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 2\begin{bmatrix} 1/2 & 0 & -1/2 \\ 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \end{bmatrix} + [0]$$

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

d) rank = 2

range = plane defined by $\underline{e}^1 + \underline{e}^2$; the vector normal to this plane is $\underline{e}^3 = \underline{e}^1 \times \underline{e}^2$

null space = $\underline{e}^3$

3 Consider the following matrix:

```
[A] =   [ 1. -1.  0.  0.  0.]
        [-1.  2. -1.  0.  0.]
        [ 0. -1.  2. -1.  0.]
        [ 0.  0. -1.  2. -1.]
        [ 0.  0.  0. -1.  2.]
```

(a) Obtain the eigensystem

(i) The diagonal components of $[\lambda]$ are the eigenvalues, and the columns of $[M^o]$ are the associated eigenvectors, together they form the eigensystem shown below.

```
[lbda]= [ 3.683  0.     0.     0.     0.   ]
        [ 0.     2.831  0.     0.     0.   ]
        [ 0.     0.     0.081  0.     0.   ]
        [ 0.     0.     0.     1.715  0.   ]
        [ 0.     0.     0.     0.     0.69 ]

 [M] =  [-0.17  -0.326 -0.597  0.456 -0.549]
        [ 0.456  0.597 -0.549 -0.326 -0.17 ]
        [-0.597 -0.17  -0.456 -0.549  0.326]
        [ 0.549 -0.456 -0.326  0.17   0.597]
        [-0.326  0.549 -0.17   0.597  0.456]
```

(ii)

$[M^o]^T[M^o] =$

```
[  1.000e+00   5.013e-16   8.546e-17  -3.753e-16  -2.776e-17]
[  5.013e-16   1.000e+00   4.905e-16  -2.285e-16   1.943e-16]
[  8.546e-17   4.905e-16   1.000e+00  -3.905e-16   3.053e-16]
[ -3.753e-16  -2.285e-16  -3.905e-16   1.000e+00  -5.551e-17]
[ -2.776e-17   1.943e-16   3.053e-16  -5.551e-17   1.000e+00]
```

(iii)

$[A^*] = [M^o][\lambda][M^o]^T =$

```
[  1.000e+00  -1.000e+00   7.494e-16  -1.110e-15  -8.327e-17]
[ -1.000e+00   2.000e+00  -1.000e+00  -8.743e-16   1.270e-15]
[  7.494e-16  -1.000e+00   2.000e+00  -1.000e+00   1.457e-15]
[ -1.055e-15  -8.743e-16  -1.000e+00   2.000e+00  -1.000e+00]
[ -1.943e-16   1.270e-15   1.402e-15  -1.000e+00   2.000e+00]
```

(iv)

$[A^*]^{-1} = [M^o][\lambda]^{-1}[M^o]^T =$

```
[ 5.  4.  3.  2.  1.]
[ 4.  4.  3.  2.  1.]
[ 3.  3.  3.  2.  1.]
[ 2.  2.  2.  2.  1.]
[ 1.  1.  1.  1.  1.]
```

(v)

$[A^*]^{-1}[A] =$

```
[  1.000e+00  -3.469e-17   6.509e-15  -1.166e-14   7.772e-15]
[ -9.437e-16   1.000e+00   6.287e-15  -1.177e-14   8.660e-15]
[ -1.388e-15   2.186e-15   1.000e+00  -9.437e-15   7.772e-15]
[ -8.049e-16  -2.567e-16   2.734e-15   1.000e+00   4.663e-15]
[ -7.216e-16   9.506e-16   1.735e-15  -3.664e-15   1.000e+00]
```

(b) Use Gershgorin's theorem to obtain bounds on the eigenvalues of [A]
The following script was written to calculate the Gershgorin circles.

```
nrow = A.shape[0]
ncol = A.shape[1]
center = np.diagonal(A)
radius = np.zeros(ncol)

for i in xrange(nrow):
    for j in xrange(ncol):
        if i != j:
            radius[i] = radius[i] + np.abs(A[i,j])
```

Based on this analysis, the eigenvalues of [A] are $0 \leq \lambda_i \leq 4$

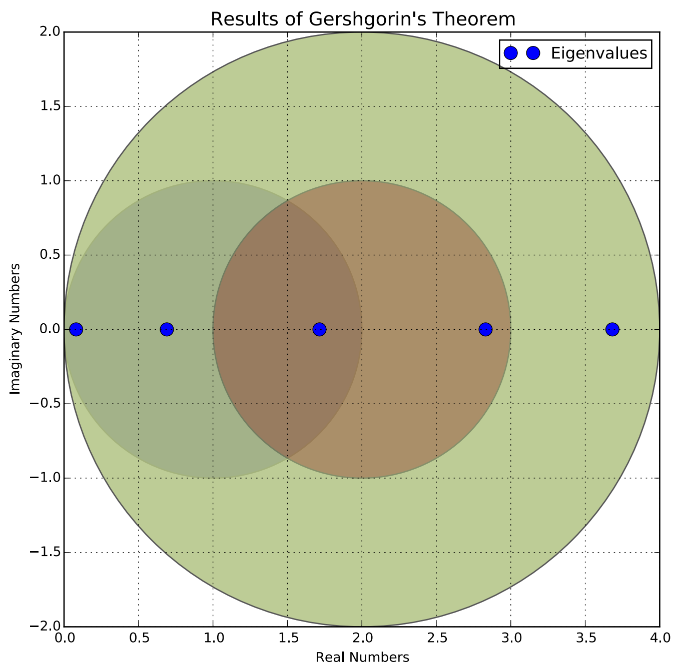


Figure 1: Gershgorin circles showing the bounds of eigenvalues for [A]..

(c) Obtain the Rayleigh quotient R(v) for $< v > = < 1, 2, 3, 2, 1 >$. Is the inequality $\lambda_1 \leq R \leq \lambda_n$ satisfied

The Rayleigh quotient R(v) = 0.263, and yes the inequality is satisfied.

(d) Obtain the norms: let a_j = the columns of [A], then

$$\|[A]\|_1 = \left(\sum_{j=1}^{n} |a_j| \right)^{1/1} = 4.0$$

$$\|[A]\|_2 = \left(\sum_{j=1}^{n} |a_j|^2 \right)^{1/2} = 3.7$$

$$\|[A]\|_\infty = max(|a_{ij}|) = 2$$

The $\|[A]\|_2$ forms an upper bound to the maximum eigenvalue.

(e) What is the condition number of [A]? The condtion number of [A] is $\frac{\lambda_{max}}{\lambda_{min}} = 45.5$

(f) Pick a solution x for the linear algebraic problem and calculate the solutions using different mode approximations.

Letting $\{x\} =< 1, 2, 3, 4, 5 >$ resulted in $\{b\} =< -1, 0, 0, 0, 6 >$, and using the definition of the spectral decomposition of [A]:

$$[A]^{-1} = \sum_{i=1}^{n} \lambda_i^{-1} \{p_i\} < p_i >$$

approximations were made by letting n range from 1 to 5 and solving

$$\{x_{ap}\} = [A]^{-1}\{b\}$$

. Approximate solutions for x are shown below, where each of the columns (one to five) is the respective approximation (one-mode, two-mode, etc.). The associated error with each of the approximations is shown below also, the five-mode approximation resulted in the smallest error by far.

```
{x_ap} = [ 0.082  -0.334   2.778   3.609   1.    ]
         [-0.221   0.542   3.402   2.808   2.    ]
         [ 0.29    0.072   2.449   1.449   3.    ]
         [-0.266  -0.848   0.852   1.161   4.    ]
         [ 0.158   0.859   1.745   2.833   5.    ]

error = [9.979e-01   9.829e-01   6.867e-01   6.413e-01   7.400e-15]
```

4 Hilbert matrices

Hilbert matrices ranging in size of n = 2,3,5,7,9,11,13,15 were analyzed, and the print out of (a) through (c) for each of these matrices is shown below.

For the linear algabraic problem, the solution $\{x\} =< 1, 2, 3, 4, 5 >$ was again chosen, which resulted in $\{b\} =< -1, 0, 0, 0, 6 >$. Figure 2 displays how the error in the linear algebraic problem increases with condition number... until it blows up (unsure why).

```
Size of Hilbert Matix: 2x2
Eigenvalues:
[ 1.268  0.066]
Condition Number
1.928147e+01
approximate solution and error
[ -6.752e-15   1.000e+00] error: 1.57039437051e-15
----------------------------------------------------------------------
Size of Hilbert Matix: 3x3
Eigenvalues:
[ 1.408  0.122  0.003]
Condition Number
5.240568e+02
approximate solution and error
[ -1.168e-12   1.000e+00   2.000e+00] error: 1.56539504238e-14
----------------------------------------------------------------------
Size of Hilbert Matix: 5x5
Eigenvalues:
[  1.567e+00   2.085e-01   1.141e-02   3.059e-04   3.288e-06]
Condition Number
4.766073e+05
approximate solution and error
[ -1.353e-06   1.000e+00   2.000e+00   3.000e+00   4.000e+00]
```

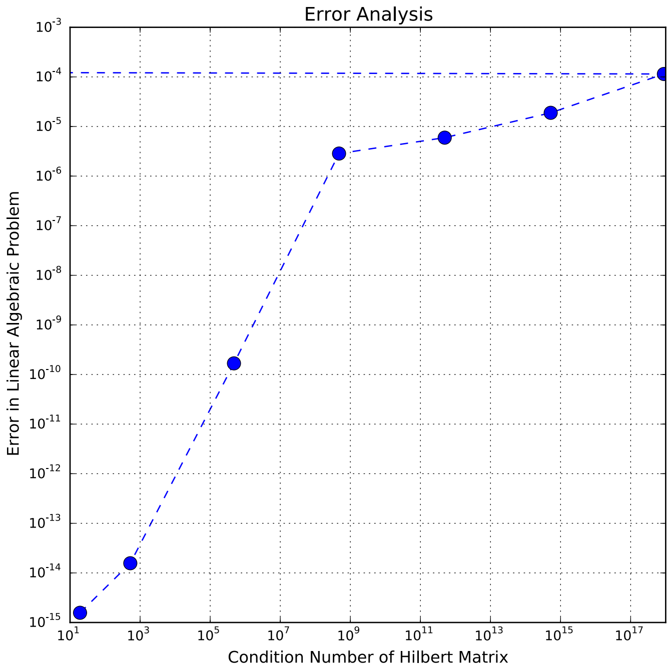


Figure 2: Plot of condtions number versus error in the solution to the linear algebraic problem..

```
error: 1.67635118738e-10
----------------------------------------------------------------------
Size of Hilbert Matix: 7x7
Eigenvalues:
[  1.661e+00   2.719e-01   2.129e-02   1.009e-03   2.939e-05   4.857e-07
   3.494e-09]
Condition Number
4.753674e+08
approximate solution and error
[  1.096e+00  -4.410e+01   4.460e+02  -1.751e+03   3.258e+03  -2.834e+03
   9.455e+02] error: 2.84920789322e-06
----------------------------------------------------------------------
Size of Hilbert Matix: 9x9
Eigenvalues:
[  1.726e+00   3.216e-01   3.104e-02   1.979e-03   8.758e-05   2.673e-06
   5.386e-08   6.461e-10   3.500e-12]
Condition Number
4.931537e+11
approximate solution and error
[  1.602e+00  -6.010e+01   5.751e+02  -2.201e+03   4.080e+03  -3.664e+03
   1.378e+03  -6.298e+01  -9.741e+00] error: 5.95702344812e-06
----------------------------------------------------------------------
Size of Hilbert Matix: 11x11
Eigenvalues:
[  1.775e+00   3.624e-01   4.031e-02   3.114e-03   1.774e-04   7.542e-06
   2.372e-07   5.368e-09   8.283e-11   7.807e-13   3.399e-15]
Condition Number
5.221964e+14
```

```
approximate solution and error
[  9.610e+00  -3.522e+02   3.237e+03  -1.227e+04   2.260e+04  -2.046e+04
   7.845e+03  -4.589e+02  -6.220e+01  -1.788e+01  -7.197e+00]
   error: 4.43459592127e-05
----------------------------------------------------------------------
Size of Hilbert Matix: 13x13
Eigenvalues:
[  1.814e+00   3.968e-01   4.903e-02   4.349e-03   2.952e-04   1.562e-05
   6.466e-07   2.076e-08   5.077e-10   9.141e-12   1.144e-13   8.892e-16
   2.042e-18]
Condition Number
8.883830e+17
approximate solution and error
[  3.277e+01  -1.278e+03   1.276e+04  -5.431e+04   1.171e+05  -1.337e+05
   7.600e+04  -1.607e+04  -3.769e+02  -8.054e+01  -2.808e+01  -1.262e+01
  -6.651e+00] error: 0.000114437392935
----------------------------------------------------------------------
Size of Hilbert Matix: 15x15
Eigenvalues:
[  1.846e+00   4.266e-01   5.721e-02   5.640e-03   4.365e-04   2.711e-05
   1.362e-06   5.529e-08   1.803e-09   4.658e-11   9.322e-13   1.394e-14
   1.421e-16   8.051e-18  -1.052e-17]
Condition Number
-1.754630e+17
approximate solution and error
[  1.195e+02  -4.681e+03   4.717e+04  -2.040e+05   4.496e+05  -5.279e+05
   3.122e+05  -7.090e+04  -1.103e+03  -2.295e+02  -7.889e+01  -3.508e+01
  -1.833e+01  -1.071e+01  -6.790e+00] error: 0.00036353130919
----------------------------------------------------------------------
```

HW04

November 3, 2015

0.1 ME 500 - Assignment 4 - Brandon Lampe

```
In [3]: from scipy import linalg as LA
        from scipy.sparse import diags as diags
        import numpy as np
        import scipy as sp
        from matplotlib import pyplot as plt

        import sys
        sys.path.append('/Users/Lampe/PyScripts')
        import blfunc as bl

        import ipdb

        np.set_printoptions(precision=3, suppress=False) # precision for numpy operations
        %precision 3
        %matplotlib inline
```

0.2 Problem 3

```
In [4]: A = diags([-1,2,-1],[-1,0,1], shape=(5,5)).toarray()
        A[0,0] = 1
        print A
```

```
[[ 1. -1.  0.  0.  0.]
 [-1.  2. -1.  0.  0.]
 [ 0. -1.  2. -1.  0.]
 [ 0.  0. -1.  2. -1.]
 [ 0.  0.  0. -1.  2.]]
```

0.2.1 3 (a)

Finde the eigenvalues and eigenvectors of $[A]$

```
In [5]: eig, Mo = LA.eig(A)
        eig_min = np.real(min(eig))
        eig_max = np.real(max(eig))
```

```
In [6]: eig_diag = eig * np.eye(5)
```

check to ensure $\Big[[A] - \lambda_1[I]\Big]\{e^1\} = 0$

```
In [7]: print (A - eig[0]*np.eye(5)).dot(Mo[:,0])
```

```
[  9.992e-16+0.j   1.554e-15+0.j   5.551e-16+0.j   1.665e-15+0.j
  -3.331e-16+0.j]
```

(i)

```
In [8]: print np.real(eig_diag)
```

```
[[ 3.683  0.     0.     0.     0.   ]
 [ 0.     2.831  0.     0.     0.   ]
 [ 0.     0.     0.081  0.     0.   ]
 [ 0.     0.     0.     1.715  0.   ]
 [ 0.     0.     0.     0.     0.69 ]]
```

```
In [33]: print Mo
```

```
[[-0.17  -0.326 -0.597  0.456 -0.549]
 [ 0.456  0.597 -0.549 -0.326 -0.17 ]
 [-0.597 -0.17  -0.456 -0.549  0.326]
 [ 0.549 -0.456 -0.326  0.17   0.597]
 [-0.326  0.549 -0.17   0.597  0.456]]
```

(ii)

```
In [34]: MoT = np.transpose(Mo)
         print Mo.dot(MoT)
```

```
[[  1.000e+00   5.013e-16   8.546e-17  -3.753e-16  -2.776e-17]
 [  5.013e-16   1.000e+00   4.905e-16  -2.285e-16   1.943e-16]
 [  8.546e-17   4.905e-16   1.000e+00  -3.905e-16   3.053e-16]
 [ -3.753e-16  -2.285e-16  -3.905e-16   1.000e+00  -5.551e-17]
 [ -2.776e-17   1.943e-16   3.053e-16  -5.551e-17   1.000e+00]]
```

(iii)

```
In [35]: A_star = Mo.dot(eig_diag).dot(MoT)
         print A_star
```

```
[[  1.000e+00+0.j  -1.000e+00+0.j   7.494e-16+0.j  -1.110e-15+0.j
   -8.327e-17+0.j]
 [ -1.000e+00+0.j   2.000e+00+0.j  -1.000e+00+0.j  -8.743e-16+0.j
    1.270e-15+0.j]
 [  7.494e-16+0.j  -1.000e+00+0.j   2.000e+00+0.j  -1.000e+00+0.j
    1.457e-15+0.j]
 [ -1.055e-15+0.j  -8.743e-16+0.j  -1.000e+00+0.j   2.000e+00+0.j
   -1.000e+00+0.j]
 [ -1.943e-16+0.j   1.270e-15+0.j   1.402e-15+0.j  -1.000e+00+0.j
    2.000e+00+0.j]]
```

(iv)

```
In [36]: A_star_inv = Mo.dot(LA.inv(eig_diag)).dot(MoT)
         print A_star_inv
```

```
[[ 5.+0.j  4.+0.j  3.+0.j  2.+0.j  1.+0.j]
 [ 4.+0.j  4.+0.j  3.+0.j  2.+0.j  1.+0.j]
 [ 3.+0.j  3.+0.j  3.+0.j  2.+0.j  1.+0.j]
 [ 2.+0.j  2.+0.j  2.+0.j  2.+0.j  1.+0.j]
 [ 1.+0.j  1.+0.j  1.+0.j  1.+0.j  1.+0.j]]
```

(v)

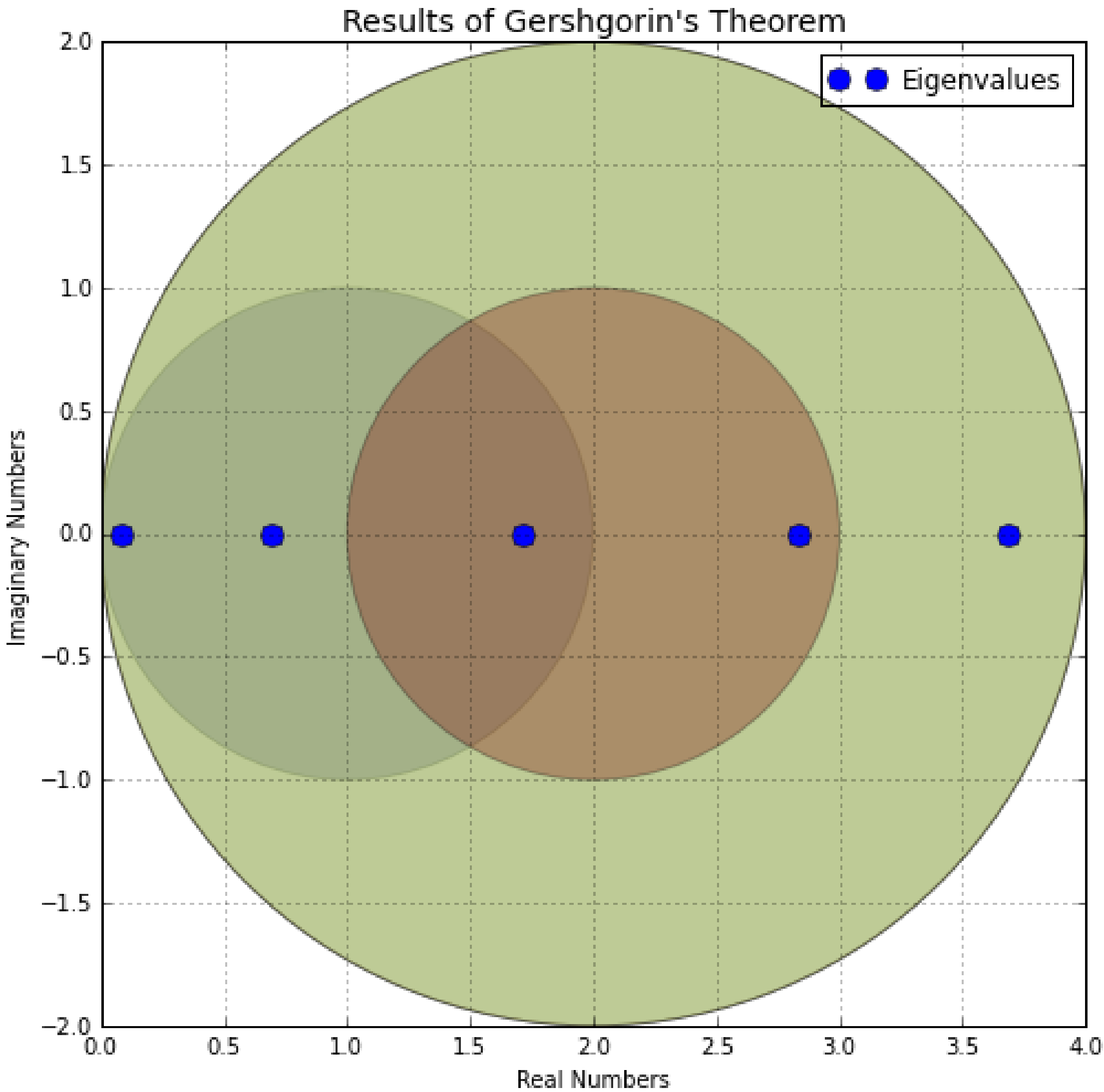
In [37]:
```
print A_star_inv.dot(A_star)
```

```
[[  1.000e+00+0.j  -3.469e-17+0.j   6.509e-15+0.j  -1.166e-14+0.j
    7.772e-15+0.j]
 [ -9.437e-16+0.j   1.000e+00+0.j   6.287e-15+0.j  -1.177e-14+0.j
    8.660e-15+0.j]
 [ -1.388e-15+0.j   2.186e-15+0.j   1.000e+00+0.j  -9.437e-15+0.j
    7.772e-15+0.j]
 [ -8.049e-16+0.j  -2.567e-16+0.j   2.734e-15+0.j   1.000e+00+0.j
    4.663e-15+0.j]
 [ -7.216e-16+0.j   9.506e-16+0.j   1.735e-15+0.j  -3.664e-15+0.j
    1.000e+00+0.j]]
```

0.2.2 3 (b)

Gersgorin's theorem to obtain bounds on eigenvalues of $[A]$

In [38]:
```
nrow = A.shape[0]
ncol = A.shape[1]
center = np.diagonal(A)
radius = np.zeros(ncol)

for i in xrange(nrow):
    for j in xrange(ncol):
        if i != j:
            radius[i] = radius[i] + np.abs(A[i,j])

fig_gersh, ax = plt.subplots(figsize = (8,8))
ax.plot(eig, np.zeros(nrow), 'o',markersize = 10, label="Eigenvalues")

ax.legend(loc=0); # upper left corner
ax.set_xlabel('Real Numbers')
ax.set_ylabel('Imaginary Numbers')
ax.set_title('Results of Gershgorin\'s Theorem' , fontsize = 14)
ax.grid(b = True, which = 'minor')
ax.grid(b = True, which = 'major')
ax.set_ylim(-2, 2)
ax.set_xlim(0,4)

bl.circles(x=center, y=np.zeros(nrow), s=radius, c=np.arange(nrow), ax=ax, alpha=0.3)

fig_name = 'plot_3b.pdf'
path = '/Users/Lampe/Documents/UNM_Courses/ME-500/HW04/'
fig_gersh.savefig(path + fig_name)
# show()
```



0.2.3 3 (c)

Obtain the Rayleigh quotient for $< v >=< 1, 2, 3, 2, 1 >$

```
In [39]: v = np.array([1,2,3,2,1])
```

```
In [40]: num = v.dot(A).dot(v)
         den = v.dot(v)
         R = num/den
         R
```

```
Out[40]: 0.263
```

0.2.4 3 (d)

Compute Norms

```
In [41]: p1 = np.linalg.norm(A, ord = 1)
         p2 = np.linalg.norm(A, ord = 2)
         pInf = np.linalg.norm(A, ord = np.inf)
         print '%.3e' %p1
         print '%.3e' %p2
         print '%.3e' %pInf
```

```
4.000e+00
3.683e+00
4.000e+00
```

```
In [42]: print '%.3e' %np.real(eig_min)
         print '%.3e' %np.real(eig_max)
```

```
8.101e-02
3.683e+00
```

0.2.5 3 (e)

The condition number of [A]

```
In [43]: cond = eig_max / eig_min
         print '%.3e' %cond
```

```
4.546e+01
```

0.2.6 3 (f)

Obtain a 1, 2, and 3-mode solution

```
In [44]: x_ex = np.array([1,2,3,4,5])
         b_ex = A.dot(x_ex)
         print b_ex
```

```
[-1.  0.  0.  0.  6.]
```

approximations for $\{x^{ex}\}$ using all 5 modes of [A]

```
In [45]: nrow = eig.shape[0]
         eig_vect_arr = np.zeros((nrow,nrow))
         eig_val_arr = np.zeros(nrow)
         A_inv_ap = np.zeros((nrow,nrow))
         x_ap = np.zeros((nrow,nrow))
         error = np.zeros(nrow)

         for i in xrange(nrow):
             A_inv_ap = 1./np.real(eig[i]) * np.outer(Mo[:,i],Mo[:,i]) + A_inv_ap
             x_ap[:,i] = A_inv_ap.dot(b_ex)
             error[i] = np.linalg.norm(x_ap[:,i] - x_ex,2)/np.linalg.norm(x_ex,2)

         print x_ap
         print error
```

```
[[ 0.082 -0.334  2.778  3.609  1.   ]
 [-0.221  0.542  3.402  2.808  2.   ]
 [ 0.29   0.072  2.449  1.449  3.   ]
 [-0.266 -0.848  0.852  1.161  4.   ]
 [ 0.158  0.859  1.745  2.833  5.   ]]
[  9.979e-01   9.829e-01   6.867e-01   6.413e-01   7.400e-15]
```

Gram-Schmidt method for obtaining eigenpairs

```
In [46]: #G-S for eigenpairs
         eig_vect_old = np.array([2,2,2,2,5])
         eig_val_old = eig_vect_old.dot(A).dot(eig_vect_old)/(eig_vect_old.dot(eig_vect_old))**0.5
         A_inv_ap_old = np.zeros((5,5))
         x_ap_arr = np.zeros((5,5))

         eig_vect_arr = np.zeros((5,5))
         eig_val_arr = np.zeros(5)
         alpha_arr = np.zeros(5)

         error_vect = np.zeros(5)
         neg_terms = np.zeros(5)

         tol = 0.001
         iter_max = 100
         error = 10
         mode_max = x_ex.shape[0]

         for h in xrange(mode_max):
             if h == 0:
                 for i in xrange(iter_max): # obtain lowest eigenpair by reverse iteration
                     x_star = bl.QR_solve(A,eig_vect_old)
                     eig_vect_new = x_star / np.sqrt(x_star.dot(x_star))
                     eig_val_new = (eig_vect_new.dot(A).dot(eig_vect_new))/den
                     error = np.abs(eig_val_new - eig_val_old)/np.abs(eig_val_new)

                     beta = eig_vect_new.dot(b_ex)
                     x_ap = beta/eig_val_new * eig_vect_new

                     eig_vect_old = eig_vect_new
                     eig_val_old = eig_val_new

                     if error <= tol:
                         eig_vect_arr[:,h] = eig_vect_new
                         eig_val_arr[h] = eig_val_new
                         break

             else:
                 for i in xrange(iter_max):
                     x_hat = bl.QR_solve(A,eig_vect_old) # x hat
                     alpha_arr[h-1] = eig_vect_arr[:,h-1].dot(x_hat)

                     neg_terms = 0
                     for j in xrange(mode_max):
                         neg_terms = neg_terms + alpha_arr[j]*eig_vect_arr[:,j]
         #                 print neg_terms

                     x_star = x_hat - neg_terms # apply G-S, corrected for previous eigenvectors
                     eig_vect_new = x_star / np.sqrt(x_star.dot(x_star))
                     den = eig_vect_new.dot(eig_vect_new) # D
                     eig_val_new = (eig_vect_new.dot(A).dot(eig_vect_new))/den
                     error = np.abs(eig_val_new - eig_val_old)/np.abs(eig_val_new)
```

```
                eig_vect_old = eig_vect_new
                eig_val_old = eig_val_new

                if error <= tol:
                    eig_vect_arr[:,h] = eig_vect_new
                    eig_val_arr[h] = eig_val_new
                    break

            for k in xrange(h+1):
                beta = eig_vect_arr[:,k].dot(b_ex)
                x_ap = beta/eig_val_arr[k] * eig_vect_arr[:,k]
                x_ap_arr[:,h] = x_ap_arr[:,h] + x_ap

            error_vect[h] = np.linalg.norm(A.dot(x_ap_arr[:,h]) - b_ex,2)/np.linalg.norm(b_ex,2)

        print eig_vect_arr
        print eig_val_arr
        print x_ap_arr
        print error_vect
```

```
[[ 0.596  0.544 -0.596 -0.556  0.593]
 [ 0.548  0.173 -0.548 -0.172  0.547]
 [ 0.456 -0.32  -0.456  0.325  0.458]
 [ 0.327 -0.598 -0.327  0.593  0.33 ]
 [ 0.17  -0.462 -0.17   0.451  0.173]]
[ 0.004  0.69   0.081  0.69   0.081]
[[ 59.545  56.928  60.057  57.43   60.693]
 [ 54.748  53.914  56.792  55.977  58.988]
 [ 45.527  47.064  49.457  50.995  53.515]
 [ 32.605  35.479  37.192  39.996  41.813]
 [ 17.008  19.23   20.123  22.254  23.207]]
[ 1.603  1.508  1.567  1.663  1.724]
```

0.2.7 4

Analysis of Hilbert Matrices

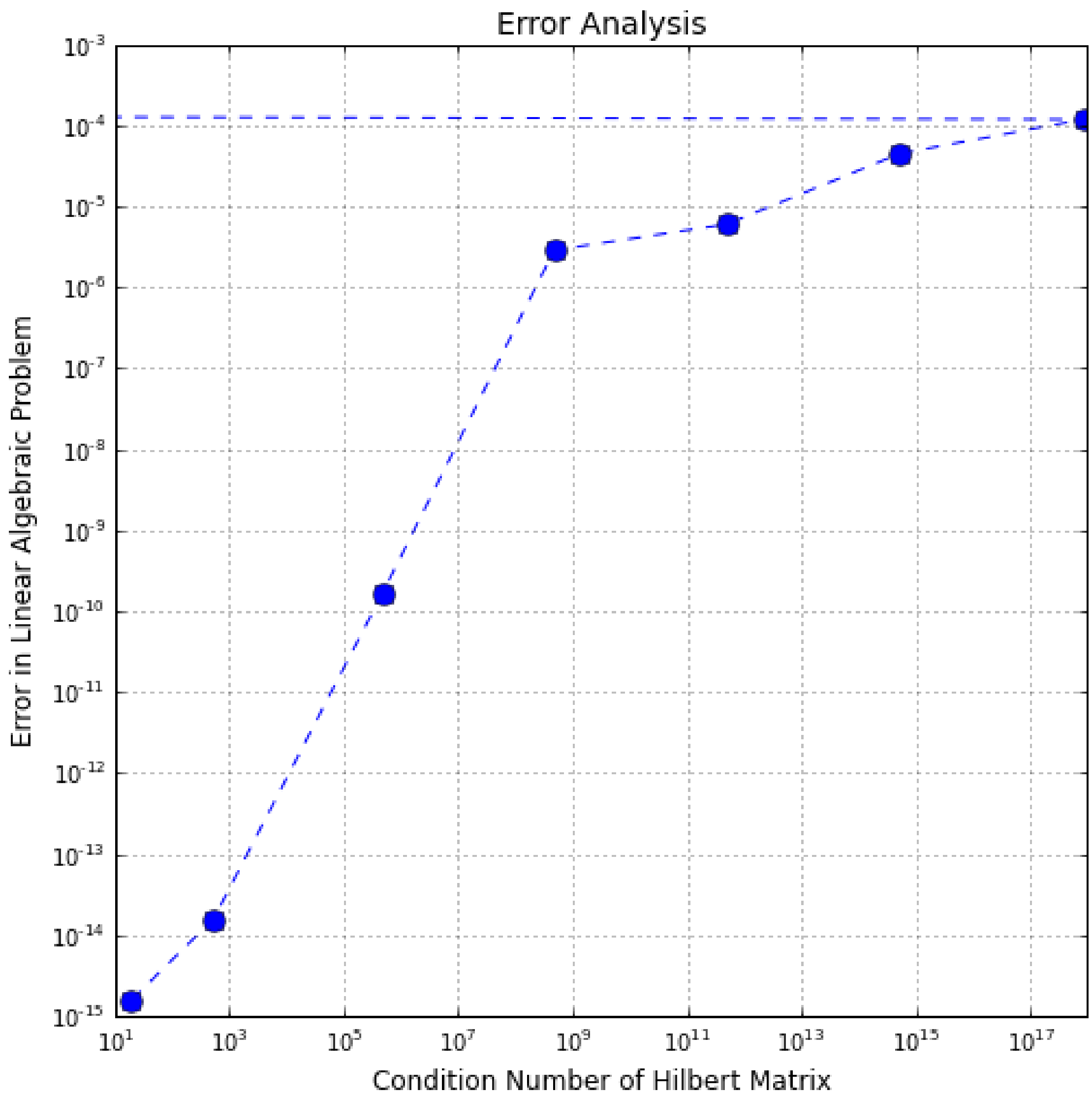
```
In [98]: rng = np.array((2,3,5,7,9,11,13,15))
         analysis = np.zeros((rng.shape[0], 2))
         inc = 0

         for i in rng:
             H = LA.hilbert(i)
             val, Mo = LA.eig(H)
             text = "Size of Hilber Matix: " + str(i)
             print text
             print "Eigenvalues:"
             print np.real(val)
             c = max(np.real(val))/min(np.real(val))
             print 'Condition Number'
             print '%e' %c
             x_ex = np.arange(i)
             b = H.dot(x_ex)
             x_ap = bl.QR_solve(H,b)
```

```
            error = np.linalg.norm(H.dot(x_ap) - b)/np.linalg.norm(b)
            print "approximate solution and error"
            out = str(x_ap) + " error: " + str(error)
            print out
            print
            analysis[inc,0] = c
            analysis[inc,1] = error
            inc = inc + 1
```

```
Size of Hilber Matix: 2
Eigenvalues:
[ 1.268  0.066]
Condition Number
1.928147e+01
approximate solution and error
[ -6.752e-15   1.000e+00] error: 1.57039437051e-15

Size of Hilber Matix: 3
Eigenvalues:
[ 1.408  0.122  0.003]
Condition Number
5.240568e+02
approximate solution and error
[ -1.168e-12   1.000e+00   2.000e+00] error: 1.56539504238e-14

Size of Hilber Matix: 5
Eigenvalues:
[  1.567e+00   2.085e-01   1.141e-02   3.059e-04   3.288e-06]
Condition Number
4.766073e+05
approximate solution and error
[ -1.353e-06   1.000e+00   2.000e+00   3.000e+00   4.000e+00] error: 1.67635118738e-10

Size of Hilber Matix: 7
Eigenvalues:
[  1.661e+00   2.719e-01   2.129e-02   1.009e-03   2.939e-05   4.857e-07
   3.494e-09]
Condition Number
4.753674e+08
approximate solution and error
[  1.096e+00  -4.410e+01   4.460e+02  -1.751e+03   3.258e+03  -2.834e+03
   9.455e+02] error: 2.84920789322e-06

Size of Hilber Matix: 9
Eigenvalues:
[  1.726e+00   3.216e-01   3.104e-02   1.979e-03   8.758e-05   2.673e-06
   5.386e-08   6.461e-10   3.500e-12]
Condition Number
4.931537e+11
approximate solution and error
[  1.602e+00  -6.010e+01   5.751e+02  -2.201e+03   4.080e+03  -3.664e+03
   1.378e+03  -6.298e+01  -9.741e+00] error: 5.95702344812e-06

Size of Hilber Matix: 11
```

```
Eigenvalues:
[  1.775e+00   3.624e-01   4.031e-02   3.114e-03   1.774e-04   7.542e-06
   2.372e-07   5.368e-09   8.283e-11   7.807e-13   3.399e-15]
Condition Number
5.221964e+14
approximate solution and error
[  9.610e+00  -3.522e+02   3.237e+03  -1.227e+04   2.260e+04  -2.046e+04
   7.845e+03  -4.589e+02  -6.220e+01  -1.788e+01  -7.197e+00] error: 4.43459592315e-05

Size of Hilber Matix: 13
Eigenvalues:
[  1.814e+00   3.968e-01   4.903e-02   4.349e-03   2.952e-04   1.562e-05
   6.466e-07   2.076e-08   5.077e-10   9.141e-12   1.144e-13   8.892e-16
   2.042e-18]
Condition Number
8.883830e+17
approximate solution and error
[  3.078e+01  -1.174e+03   1.144e+04  -4.740e+04   9.904e+04  -1.087e+05
   5.863e+04  -1.129e+04  -3.642e+02  -7.949e+01  -2.795e+01  -1.262e+01
  -6.671e+00] error: 0.000118244924399

Size of Hilber Matix: 15
Eigenvalues:
[  1.846e+00   4.266e-01   5.721e-02   5.640e-03   4.365e-04   2.711e-05
   1.362e-06   5.529e-08   1.803e-09   4.658e-11   9.322e-13   1.394e-14
   1.421e-16   8.051e-18  -1.052e-17]
Condition Number
-1.754630e+17
approximate solution and error
[  1.188e+02  -4.647e+03   4.676e+04  -2.020e+05   4.444e+05  -5.208e+05
   3.074e+05  -6.961e+04  -1.102e+03  -2.294e+02  -7.887e+01  -3.508e+01
  -1.834e+01  -1.071e+01  -6.791e+00] error: 0.000364001386973
```

```
In [99]: fig_Hilb, ax = plt.subplots(figsize = (8,8))

         ax.plot(analysis[:,0],analysis[:,1], 'o--', markersize=10)

         # ax.legend(loc=0); # upper left corner
         ax.set_xlabel('Condition Number of Hilbert Matrix', fontsize = 12)
         ax.set_ylabel('Error in Linear Algebraic Problem', fontsize = 12)
         ax.set_title('Error Analysis' , fontsize = 14)
         ax.grid(b = True, which = 'major')
         ax.grid(b = True, which = 'major')
         # ax.set_ylim(-2, 2)
         # ax.set_xlim(0,4)
         ax.set_xscale('log')
         ax.set_yscale('log')

         fig_name = 'plot_4.pdf'
         path = '/Users/Lampe/Documents/UNM_Courses/ME-500/HW04/'
         fig_gersh.savefig(path + fig_name)
         # show()
```


Error Analysis
Error in Linear Algebraic Problem
Condition Number of Hilbert Matrix
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
10^{-11}
10^{-12}
10^{-13}
10^{-14}
10^{-15}
10^{1}
10^{3}
10^{5}
10^{7}
10^{9}
10^{11}
10^{13}
10^{15}
10^{17}

```
In [ ]:
```