Numerical Linear Algebra: Solutions for Sheet 2

1. Consider the least squares problem $\min_{x\in\mathbb{R}^n} \|Ax-b\|_2$, where $A\in\mathbb{R}^{m\times n}$ with $m\geq n$, $b\in\mathbb{R}^m$, and we assume that $\operatorname{rank}(A)=n$. We can then write $A=QR$, where $Q\in\mathbb{R}^{m\times m}$ is orthogonal and $R\in\mathbb{R}^{m\times n}$ is upper triangular. Then our least squares problem can be rewritten as

$$\min_{x\in\mathbb{R}^n}\|Ax-b\|_2=\min_{x\in\mathbb{R}^n}\|QRx-b\|_2=\min_{x\in\mathbb{R}^n}\|QRx-QQ^{\mathrm{T}}b\|_2=\min_{x\in\mathbb{R}^n}\|Rx-Q^{\mathrm{T}}b\|_2,$$

where the last step follows from the orthogonal invariance of the 2-norm. Now note that the matrix $R\in\mathbb{R}^{m\times n}$ and the vector $w\equiv Q^{\mathrm{T}}b\in\mathbb{R}^m$ can be partitioned in the form

$$R=\begin{pmatrix}\widehat{R}\\0\end{pmatrix}\qquad w=\begin{pmatrix}\widehat{w}\\\widetilde{w}\end{pmatrix}\qquad\text{where }\widehat{R}\in\mathbb{R}^{n\times n},\ \widehat{w}\in\mathbb{R}^n,\text{ and }\widetilde{w}\in\mathbb{R}^{m-n}.\tag{1}$$

Note in particular that, since $\operatorname{rank}(A)=n$, $\widehat{R}$ is non-singular. Thus, the solution of the least squares problem is $x=\widehat{R}^{-1}\widehat{w}$, and $\min_{x\in\mathbb{R}^n}\|Ax-b\|_2=\|\widetilde{w}\|_2$.

Now consider the the normal equations, $A^{\mathrm{T}}Ay=A^{\mathrm{T}}b$. We wish to show that y is identical to the x obtained above. Since $\operatorname{rank}(A)=n$, $A^{\mathrm{T}}A\in\mathbb{R}^{n\times n}$ is non-singular. (Think about how you would prove this using the SVD; see Question 7 of Sheet 1.) As before, let $A=QR$ be a QR factorization of A. Since $A^{\mathrm{T}}A$ is invertible, we can write $y=(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b=(R^{\mathrm{T}}Q^{\mathrm{T}}QR)^{-1}R^{\mathrm{T}}Q^{\mathrm{T}}b=(R^{\mathrm{T}}R)^{-1}R^{\mathrm{T}}Q^{\mathrm{T}}b$. Breaking R and $w\equiv Q^{\mathrm{T}}b$ does as we did in (1) above, note that $(R^{\mathrm{T}}R)^{-1}=(\widehat{R}^{\mathrm{T}}\widehat{R})^{-1}=\widehat{R}^{-1}\widehat{R}^{-\mathrm{T}}$, since $\widehat{R}$ is a square, non-singular matrix, and also $R^{\mathrm{T}}w=\widehat{R}^{\mathrm{T}}\widehat{w}$. Thus, we have $y=(R^{\mathrm{T}}R)^{-1}R^{\mathrm{T}}Q^{\mathrm{T}}b=\widehat{R}^{-1}\widehat{R}^{-\mathrm{T}}\widehat{R}^{\mathrm{T}}\widehat{w}=\widehat{R}^{-1}\widehat{w}$. That is, we have the same formula for y as we obtained for x above.

Now let $A=U\Sigma V^{\mathrm{T}}$ be a singular value decomposition of A, where $U\in\mathbb{R}^{m\times m}$ and $V\in\mathbb{R}^{n\times n}$ are orthogonal and $\Sigma\in\mathbb{R}^{m\times n}$ is diagonal (with non-negative, non-increasing entries). Then we have

$$\min_{x\in\mathbb{R}^n}\|Ax-b\|_2=\min_{x\in\mathbb{R}^n}\|U\Sigma V^{\mathrm{T}}x-b\|_2=\min_{x\in\mathbb{R}^n}\|U\Sigma V^{\mathrm{T}}x-UU^{\mathrm{T}}b\|_2=\min_{x\in\mathbb{R}^n}\|\Sigma V^{\mathrm{T}}x-U^{\mathrm{T}}b\|_2,$$

where the last equality follows from the orthogonal invariance of the 2-norm. As we did in the QR context above, we can partition Σ and $s\equiv U^{\mathrm{T}}b$ in the form

$$\Sigma=\begin{pmatrix}\widehat{\Sigma}\\0\end{pmatrix}\qquad s=\begin{pmatrix}\widehat{s}\\\widetilde{s}\end{pmatrix}\qquad\text{where }\widehat{\Sigma}\in\mathbb{R}^{n\times n},\ \widehat{s}\in\mathbb{R}^n,\text{ and }\widetilde{s}\in\mathbb{R}^{m-n}.$$

Thus the solution x to the least squares problem $\min_{x\in\mathbb{R}^n}\|Ax-b\|_2$ satisfies $\widehat{\Sigma}V^{\mathrm{T}}x=\widehat{s}$. Since $\operatorname{rank}(A)=n$, A has n non-zero singular values, and $\widehat{\Sigma}$ is invertible. Thus we can write $x=V\widehat{\Sigma}^{-1}\widehat{s}$.

2. We first note that in order convert the problem into a linear approximation problem, we must take logarithms of both sides. Writing $a=e^b$ and $w=\lambda\mu$, we have

$$\log(y)=b+\lambda t-w$$

and then we can formulate the least squares problem as

$$\min_{b,\lambda,\mu}\left\|\begin{pmatrix}1&t_1&-1\\1&t_2&-1\\\vdots&\vdots&\vdots\\1&t_n&-1\end{pmatrix}\begin{pmatrix}b\\\lambda\\\mu\end{pmatrix}-\begin{pmatrix}\log(y_1)\\\log(y_2)\\\vdots\\\log(y_n)\end{pmatrix}\right\|_2\tag{2}$$

In this formulation, A, the co-efficient matrix of b, λ, μ is clearly rank deficient because a and μ perform the same function in the original model. This can similarly be seen in the QR factorization of A where

$$R = \begin{pmatrix} -2 & -3 & -2 \\ 0 & -2.2361 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Setting $w = 0$ removes the last column of A and therefore also the last column of R, and the linear least squares problem (having multiplied through by Q^T which is orthogonal and hence doesn't change the norm) becomes

$$\min_{b,\lambda} \left\| \begin{pmatrix} -2 & 3 \\ 0 & -2.2361 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} b \\ \lambda \end{pmatrix} - \begin{pmatrix} -2.3937 \\ 1.8718 \\ -0.0591 \\ 0.0911 \end{pmatrix} \right\|_2 \tag{3}$$

¿From which we can find b, λ by

$$\begin{pmatrix} b \\ \lambda \end{pmatrix} = \begin{pmatrix} -2 & 3 \\ 0 & -2.2361 \end{pmatrix}^{-1} \begin{pmatrix} -2.3937 \\ 1.8718 \end{pmatrix} = \begin{pmatrix} 2.4525 \\ -0.8371 \end{pmatrix} \tag{4}$$

with residual

$$\left\| \begin{pmatrix} -0.0591 \\ 0.0911 \end{pmatrix} \right\|_2 = 0.1086$$

So our best least squares model is $y = e^{-0.8371t+2.4525}$ with least squares error of 0.1086.

3. To demonstrate that the shifted QR algorithm does not alter the eigenvalues, recall the algorithm:

$$\begin{aligned} &QR \text{ factorization of } A_k - \mu_k I: \quad & Q_k R_k &:= (A_k - \mu_k I) \qquad (5) \\ &\text{and then form} & A_{k+1} &:= R_k Q_k + \mu_k I, \qquad (6) \end{aligned}$$

with $A_1 = A$. Clearly A_1 has the same eigenvalues as A. Now we want to prove that A_{k+1} has the same eigenvalues as A_k. The key is to show that these matrices are *similar*. Note that equation (5) gives $R_k = Q_k^{\mathrm{T}}(A_k - \mu_k I)$. We can use this expression to manipulate equation (6),

$$A_{k+1} = R_k Q_k + \mu_k I = Q_k^{\mathrm{T}}(A_k - \mu_k I)Q_k + \mu_k I = Q_k^{\mathrm{T}}(A_k - \mu_k I)Q_k + \mu_k Q_k^{\mathrm{T}} Q_k = Q_k^{\mathrm{T}} A_k Q_k.$$

Since Q_k is orthogonal, $Q_k^{\mathrm{T}} = Q_k^{-1}$, so A_{k+1} is a similarity transformation of A_k and thus has the same eigenvalues. Thus by mathematical induction, A_k has the same eigenvalues as A for any $k > 0$.

4. In MATLAB, first type `format long` to display enough digits to appreciate what is happening.

   After 10 iterations, `A` is

```
9.39361465509614   0.09663042293221  -0.25355643551638
0.00107438128642   4.70636353667089  -0.96251422635769
0.00000011415451   0.00007505211157   1.90002180823296
```

   After 48 iterations, `A` is

```
9.39363679919756   0.09554927156795  -0.25377973238817
0.00000000000000   4.70631564857751  -0.96253113411038
0.00000000000000   0.00000000000000   1.90004755222494
```

   Notice that `eig([9 1 1;1 5 0; 1 1 2])` computes

```
9.39363679919755   4.70631564857751  1.90004755222494
```

   ***QR* Movie**: Try out the following MATLAB code, which visually illustrates the convergence of the QR algorithm. We plot the logarithm of each matrix entry's magnitude: large entries are red, small entries ($\leq 10^{-16}$ in magnitude) are blue.

```
 N = 20;
 V = randn(N);                % random eigenvector matrix
 A = V*diag([1:N])*inv(V);    % A has eigenvalues 1, 2, ..., 25, random eigenvectors
%A = randn(N);                % contrast with the A above

 I = eye(N); j=0;
 x = kron([1:N+1]-.5,[1 1]); x = x(2:end-1);
 while (norm(diag(A,-1)) > 1e-14)
    [Q,R] = qr(A); A = R*Q; j=j+1;
    surf(x,x,kron(log10(abs(A)+1e-16),[1 1;1 1]));
    axis([.5 N+.5 .5 N+.5]), view(0,-90), caxis([-16,1]), colorbar
    title(sprintf('QR algorithm: iteration %d', j),'fontsize', 16);
    text(N+(N/20)*2.2,N+(N/20),'color indicates')
    text(N+(N/20)*2.7,N+(N/20)*2.1,'log_{10} | a_{ij}^{(k)} |')
    drawnow;
 end;
```

   Repeat the above experiment, but replace the third line with:

```
A = V*diag([2; ones(N-1,1)])*inv(V);
```

   How does convergence behavior change?
   Can you improve convergence using the shifting ideas of Question 1?

5. In the lecture notes, it was shown that if $\|u\|_2 = \|v\|_2$, then $H(w)u = v$ when $w = r(u - v)$ for any scalar $r \neq 0$. This case very similar. (Notice that $(u^{\mathrm{T}} H(w))^{\mathrm{T}} = H(w)u$.) Letting $w = r(u - v)$, we have $w^{\mathrm{T}} w = r^2 (u - v)^{\mathrm{T}} (u - v) = r^2 (u^{\mathrm{T}} u - 2u^{\mathrm{T}} v + v^{\mathrm{T}} v) = 2r^2 (u^{\mathrm{T}} u - u^{\mathrm{T}} v) = 2ru^{\mathrm{T}} w$, where we have used the fact that $\|u\|_2 = \|v\|_2 \implies u^{\mathrm{T}} u = v^{\mathrm{T}} v$. Thus,

$$u^{\mathrm{T}} H(w) = u^{\mathrm{T}} \left(I - 2\frac{ww^{\mathrm{T}}}{w^{\mathrm{T}} w} \right) = u^{\mathrm{T}} - \frac{2u^{\mathrm{T}} ww^{\mathrm{T}}}{2ru^{\mathrm{T}} w} = u^{\mathrm{T}} - \frac{w^{\mathrm{T}}}{r} = u^{\mathrm{T}} - (u - v)^{\mathrm{T}} = v^{\mathrm{T}}.$$

Let $A \in \mathbb{R}^{n \times n}$ and write

$$A = A_1 = \begin{pmatrix} u_1^{\mathrm{T}} \\ \widehat{A}_1 \end{pmatrix},$$

where $\widehat{A}_1 \in \mathbb{R}^{n-1 \times n}$. Then we seek the Householder reflector $H_1(w_1)$ such that $A_1 H_1(w_1)$ has zeroes in the first row, except for the (1,1) entry. To accomplish this, we set $v = \pm\|u_1\|_2 e_1$, where e_1 is the first column of the $n \times n$ identity matrix, and follow the construction from the first part of the question. This yields

$$A_1 H_1(w_1) = \left(\begin{array}{c|ccc} \alpha_1 & 0 & \cdots & 0 \\ \hline \times & & & \\ \vdots & & A_2 & \\ \times & & & \end{array} \right), \qquad \text{where } \alpha_1 = \pm\|u_1\|_2 \text{ and } A_2 \in \mathbb{R}^{n-1 \times n-1}.$$

where $\times$ reflects a generic non-zero entry. We now apply this process inductively, working on A_2 as we did on A_1. This generates an $n - 1 \times n - 1$ Householder reflector, which we'll call $K_2(w_2)$, with

$$A_2 K_2(w_2) = \left(\begin{array}{c|ccc} \alpha_2 & 0 & \cdots & 0 \\ \hline \times & & & \\ \vdots & & A_3 & \\ \times & & & \end{array} \right), \qquad \text{where } \alpha_2 = \pm\|u_2\|_2 \text{ and } A_3 \in \mathbb{R}^{n-2 \times n-2}.$$

We then set

$$H_2(w_2) = \left(\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & K_2 & \\ 0 & & & \end{array} \right) \in \mathbb{R}^{n \times n},$$

which is also an orthogonal matrix. Continuing inductively, we have $A(H_1(w_1)\, H_2(w_2) \cdots H_n(w_n)) = AQ^{\mathrm{T}} = L$, which implies $A = LQ$.

The LQ factorization can also be seen by taking the QR decomposition of A^{T}. That is, let $A^{\mathrm{T}} = QR$ for orthogonal Q and upper triangular R. Then $A = (A^{\mathrm{T}})^{\mathrm{T}} = R^{\mathrm{T}} Q^{\mathrm{T}}$. Note that R^{T} is lower triangular and Q^{T} is orthogonal. Thus, this is an LQ factorization of A.

6.

$$
J(i,k,\theta) = \begin{array}{cl}
\begin{array}{ccccccccccc} & & & \text{col. } i & & & & \text{col. } k & & & \end{array} & \\
\left(\begin{array}{ccccccccccc}
1 & & & & & & & & & & \\
 & \ddots & & & & & & & & & \\
 & & 1 & & & & & & & & \\
 & & & \cos\theta & & & & \sin\theta & & & \\
 & & & & 1 & & & & & & \\
 & & & & & \ddots & & & & & \\
 & & & & & & 1 & & & & \\
 & & & -\sin\theta & & & & \cos\theta & & & \\
 & & & & & & & & 1 & & \\
 & & & & & & & & & \ddots & \\
 & & & & & & & & & & 1
\end{array}\right) &
\begin{array}{l} \\ \\ \\ \text{row } i \\ \\ \\ \\ \text{row } k \\ \\ \\ \\ \end{array}
\end{array}
$$

Let $y = J(i,k,\theta)x$. Then we have

$$
y_j = \begin{cases} x_j & j \neq i, j \neq k; \\ \cos(\theta)x_i + \sin(\theta)x_k & j = i; \\ -\sin(\theta)x_i + \cos(\theta)x_k & j = k. \end{cases}
$$

Let $\theta = \cos^{-1}(x_i/\sqrt{x_i^2 + x_k^2})$. Then

$$
\cos(\theta) = \frac{x_i}{\sqrt{x_i^2 + x_k^2}} \qquad \text{and} \qquad \sin(\theta) = \frac{x_k}{\sqrt{x_i^2 + x_k^2}}.
$$

Then $y_k = -\sin(\theta)x_i + \cos(\theta)x_k = 0$.

Let A be upper Hessenberg. (This means the only non-zero entries are on the first subdiagonal and above; $a_{i,j} = 0$ if $i > j + 1$.) We can construct a QR factorization of A by eliminating the non-zero sub-diagonal entries from upper left to lower right. $J(1, 2, \theta_1)A$ will only alter the first two rows of A. Selecting $\theta_1 = \cos^{-1}(a_{1,1}/(a_{1,1}^2 + a_{2,1}^2)^{1/2})$ will eliminate the (2,1) entry of A, altering only the first and second row of A. Thus, $A^{(1)} := J(1, 2, \theta_1)A$ has the desired form in the first two rows. Now, we must eliminate the non-zero subdiagonal entry in the third row of $A^{(1)}$. This is done by the Givens rotation $J(2, 3, \theta_2)$, with $\theta_2 = \cos^{-1}(a_{2,2}^{(1)}/((a_{1,1}^{(1)})^2 + (a_{2,1}^{(1)})^2)^{1/2})$. Premultiplying $A^{(1)}$ by this rotation will alter the second and third row of $A^{(1)}$. Do the zero (2,1) and (3,1) entries become non-zero? No: Since the updated values are just linear combinations of zero entries, the zero structure is maintained. This process is continued inductively, giving

$$
J(n-1, n, \theta_{n-1}) \cdots J(2, 3, \theta_2)J(1, 2, \theta_1)A = R \implies A = J(1, 2, \theta_1)^{\mathrm{T}}\, J(2, 3, \theta_2)^{\mathrm{T}} \cdots J(n-1, n, \theta_{n-1})^{\mathrm{T}} R.
$$

The product of orthogonal matrices is orthogonal.

7. The Gaussian elimination algorithm yields

$$
\ell_{2,1} = a_{2,1}/a_{1,1} = 1/9 \qquad M(2,1,-\ell_{2,1})\,A = \begin{pmatrix} 1 & 0 & 0 \\ -1/9 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 9 & 1 & 1 \\ 1 & 5 & 0 \\ 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 9 & 1 & 1 \\ 0 & 44/9 & -1/9 \\ 1 & 1 & 2 \end{pmatrix} =: A^{(1)}
$$

$$
\ell_{3,1} = a_{3,1}^{(1)}/a_{1,1}^{(1)} = 1/9 \qquad M(3,1,-\ell_{3,1})\,A^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/9 & 0 & 1 \end{pmatrix} \begin{pmatrix} 9 & 1 & 1 \\ 0 & 44/9 & -1/9 \\ 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 9 & 1 & 1 \\ 0 & 44/9 & -1/9 \\ 0 & 8/9 & 17/9 \end{pmatrix} =: A^{(2)}
$$

$$
\ell_{3,2} = a_{3,2}^{(2)}/a_{2,2}^{(2)} = 2/11 \qquad M(3,2,-\ell_{3,2})\,A^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2/11 & 1 \end{pmatrix} \begin{pmatrix} 9 & 1 & 1 \\ 0 & 44/9 & -1/9 \\ 0 & 8/9 & 17/9 \end{pmatrix} = \begin{pmatrix} 9 & 1 & 1 \\ 0 & 44/9 & -1/9 \\ 0 & 0 & 21/11 \end{pmatrix}.
$$

That is,

$$
L = \begin{pmatrix} 1 & 0 & 0 \\ 1/9 & 1 & 0 \\ 1/9 & 2/11 & 1 \end{pmatrix} \qquad U = \begin{pmatrix} 9 & 1 & 1 \\ 0 & 44/9 & -1/9 \\ 0 & 0 & 21/11 \end{pmatrix}.
$$

We use this result to solve the specified linear system with two triangular solves:

$$Ly = \begin{pmatrix} 11 \\ 6 \\ 4 \end{pmatrix} \implies y = \begin{pmatrix} 11 \\ 43/9 \\ 21/11 \end{pmatrix}, \qquad Ux = y \implies x = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

8. Partial pivoting occurs during elimination of row p if $|a_{p,p}| < \max_j |a_{j,p}|$ for $j = p+1, \ldots, n$. If this occurs, then the pth row is swapped with the jth row. We modify the matrix in Question 5 to now be

$$A = \begin{pmatrix} 9 & 1 & 1 \\ 1 & 5 & 0 \\ 10 & 1 & 2 \end{pmatrix}.$$

At the first step, $9 = |a_{1,1}| < |a_{3,1}| = 10$, and thus we swap the first and third row:

$$PA = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 9 & 1 & 1 \\ 1 & 5 & 0 \\ 10 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 10 & 1 & 2 \\ 1 & 5 & 0 \\ 9 & 1 & 1 \end{pmatrix}.$$

For this example, only this one pivot is required, and we have in the end $PA = LU$, with

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ 1/10 & 1 & 0 \\ 9/10 & 1/49 & 1 \end{pmatrix} \qquad U = \begin{pmatrix} 10 & 1 & 2 \\ 0 & 49/10 & -2/10 \\ 0 & 0 & -39/49 \end{pmatrix}.$$

9. A row swap would occur at the first step if $|a_{1,1}|$ is not the largest magnitude entry in the first column. The strictly column diagonally dominant (SCDD) property ensures that it is the largest magnitude entry in the column, and thus no pivoting occurs.

After one step of Gaussian elimination, we have $B \in \mathbb{R}^{n-1 \times n-1}$. We can write a generic entry of B as

$$b_{i,j} = a_{i+1,j+1} - \frac{a_{i+1,1}}{a_{1,1}} a_{1,j+1}, \qquad i, j \in \{1, \ldots, n-1\}.$$

To prove that B is SCDD, consider

$$\sum_{i=1, i \neq j}^{n-1} |b_{i,j}| = \sum_{i=1, i \neq j}^{n-1} \left| a_{i+1,j+1} - \frac{a_{i+1,1}}{a_{1,1}} a_{1,j+1} \right| \leq \sum_{i=1, i \neq j}^{n-1} |a_{i+1,j+1}| + \frac{|a_{1,j+1}|}{|a_{1,1}|} \sum_{i=1, i \neq j}^{n-1} |a_{i+1,1}|$$

Since A is SCDD, we have

$$\sum_{i=1, i \neq j}^{n-1} |a_{i+1,j+1}| < |a_{j+1,j+1}| - |a_{1,j+1}| \qquad \text{and} \qquad \sum_{i=1, i \neq j}^{n-1} |a_{i+1,1}| < |a_{1,1}| - |a_{j+1,1}|.$$

Substituting these expressions into the bound, we have

$$\begin{aligned} \sum_{i=1, i \neq j}^{n-1} |b_{i,j}| &< |a_{j+1,j+1}| - |a_{1,j+1}| + \frac{|a_{1,j+1}|}{|a_{1,1}|} (|a_{1,1}| - |a_{j+1,1}|) = |a_{j+1,j+1}| - \frac{|a_{1,j+1}|}{|a_{1,1}|} |a_{j+1,1}| \\ &\leq \left| a_{j+1,j+1} - \frac{a_{1,j+1}}{a_{1,1}} a_{j+1,1} \right| = |b_{j,j}|. \end{aligned}$$

Thus, B is SCDD as well. Therefore, Gaussian elimination preserves the SCDD property at each step, and no partial pivoting is necessary during the LU factorization of A.

10. Suppose $Ax = b$ and $(A+\delta A)(x+\delta x) = b$, which together imply $(A+\delta A)(x+\delta x) = Ax$. Multiplying this out gives $Ax + A\delta x + \delta A(x+\delta x) = Ax$ and consequently $A\delta x = -\delta A(x+\delta x)$. Assuming A is invertible, $\delta x = -A^{-1}(\delta A)(x+\delta x)$. We now apply norm inequalities to obtain

$$\|\delta x\| = \|A^{-1}(\delta A)(x+\delta x)\| \leq \|A^{-1}\|\,\|\delta A\|\,\|x+\delta x\| = \frac{\|A\|}{\|A\|}\|A^{-1}\|\,\|\delta A\|\,\|x+\delta x\|.$$

Dividing by $\|x+\delta x\|$ gives the desired result,

$$\frac{\|\delta x\|}{\|x+\delta x\|} \leq \|A\|\,\|A^{-1}\|\frac{\|\delta A\|}{\|A\|}.$$

This bounds the relative perturbation of the solution by the relative perturbation in the matrix, magnified by the condition number $\kappa(A) = \|A\|\,\|A^{-1}\|$. Small errors in A might lead to large errors in the computed solution.