

2) a) We have to prove that $\|x^T A x\| \leq \|A\|_2$, where x is a unit vector and A is a symmetric matrix.

Using Cauchy-Schwarz theorem, $|u^T v| \leq \|u\|_2 \cdot \|v\|_2$, we know that

$$\|x^T A x\| \leq \|x^T\|_2 \cdot \|A x\|_2$$

Because $\|x\|_2 = \|x^T\|_2 = 1$, we can simplify to: $\|x^T A x\| \leq \|A x\|_2$ ①

From the definition of vector induced norm, we know that $\|A x\|_2 \leq \|A\|_2$ ②

Combining the two inequalities, ① and ②, we get:

$$\|A\|_2 \cdot \|x^T A x\| \leq \|A x\|_2 \leq \|A\|_2$$

Hence, $\|x^T A x\| \leq \|A\|_2$ (proven)

b) Because we know that $\tilde{\lambda}$ and \tilde{u} are the respective eigenvalue and eigenvector for \tilde{A} , we can say that:

$$\tilde{A} \cdot \tilde{u} = \tilde{\lambda} \tilde{u}$$

$$(A + \delta A)(u + \delta u) = (\lambda + \delta \lambda)(u + \delta u) \quad (\text{Substitute the perturbation made})$$

$$\cancel{A}u + \delta A u + A \delta u + \delta A \delta u = \cancel{\lambda}u + \delta \lambda u + \delta u + \delta(\delta u) \quad (\text{Expand})$$

$$\cancel{\delta A}u + A \delta u + \delta A \delta u = \delta \lambda u + \lambda \delta u + \cancel{\delta(\delta u)} \quad (\text{Because } \cancel{\delta A}u = \lambda \cancel{\delta u})$$

$$\delta A u + A \delta u = \delta \lambda u + \lambda \delta u \quad (\text{Because } \cancel{\delta A} \delta u \ll \epsilon_{\text{machine}} \text{ and } \delta \lambda \delta u \ll \epsilon_{\text{machine}})$$

$$u^T \delta A u + u^T A \delta u = u^T \delta \lambda u + u^T \lambda \delta u \quad (\text{Pre-multiply with } u^T)$$

$$\underbrace{u^T \delta A u}_{\sim \sim \sim} + \lambda \underbrace{u^T du}_{\sim \sim \sim} = \cancel{\lambda u^T u} + \cancel{\lambda u^T du} \quad (\text{Because } u^T \delta A u = \cancel{\lambda u^T u}, \text{ proven later})$$

$$\underbrace{u^T \delta A u}_{\sim \sim \sim} = \cancel{\lambda u^T u} \quad (\text{Subtract } \cancel{\lambda u^T du} \text{ from both sides})$$

$$\underbrace{u^T \delta A u}_{\sim \sim \sim} = \cancel{\lambda u^T u} \quad (\text{Take } u \text{ to be unit eigenvector})$$

$$|\delta \lambda| = |\underbrace{u^T \delta A u}_{\sim \sim \sim}| \quad (\text{proven})$$

$$\text{As proven in a), } |\underbrace{u^T \delta A u}_{\sim \sim \sim}| \leq \|\underbrace{\delta A}_{\sim}\|_2$$

$$\text{Because } |\delta \lambda| = |\underbrace{u^T \delta A u}_{\sim \sim \sim}| = |\delta \lambda| \leq \|\underbrace{\delta A}_{\sim}\|_2 \quad (\text{proven})$$

$$\text{Proof that } \underbrace{u^T A \delta u}_{\sim \sim \sim} = \lambda \underbrace{u^T \delta u}_{\sim \sim \sim}$$

$$\text{L.H.S.} = \underbrace{u^T \underbrace{A \delta u}_{\sim}}_{\sim}$$

$$= \underbrace{u^T A^T \delta u}_{\sim \sim} \quad (\text{Because } A \text{ is symmetric, } A = A^T)$$

$$= (\underbrace{A u}_{\sim})^T \underbrace{\delta u}_{\sim} \quad (\text{Reversal law: } A^T B^T = (BA)^T)$$

$$= (\underbrace{A u}_{\sim})^T \underbrace{\delta u}_{\sim} = (\underbrace{A u}_{\sim} = \lambda u) A^T \delta u \quad (\cancel{A^T \delta u = \lambda u})$$

(proven)

c) Let us first find out the inputs and outputs of the problem:

Input: ΔA Output: λ

We hence know that $\kappa^R \leq \frac{\|d\|_1}{\|\lambda\|}$

$$\frac{\|d\|_1}{\|\lambda\|}$$

$$\frac{\|dA\|_2}{\|A\|_2}$$

$$\frac{\|A\|_2}{\|A\|_2}$$

$$\leq \frac{|s\lambda|}{|\lambda|} \times \frac{\|A\|_2}{\|A\|_2} \quad (\text{Because } s\lambda \text{ and } \lambda \text{ are scalar, the norm can be replaced with modulus})$$

$$(\text{Because of result of b)}) \leq \frac{\|A\|_2}{\|A\|_2}$$

Hence, the condition number of this problem is $\frac{\|A\|_2}{\|\lambda\|}$.

d) From c), we know that $\kappa^R = \frac{\|A\|_2}{\|\lambda\|} \cdot \frac{\|A\|_2}{\|A\|_2}$

$$\|A\|_2 = a \quad (\text{Because } \|D\|_2 = \max_{1 \leq i \leq m} |d_{ii}| \text{ when } D \text{ is an mxm diagonal matrix})$$

$$\text{Hence, } \kappa^R = \frac{\|A\|_2}{\|\lambda\|} = \frac{a}{a} = 1$$

Because the relative condition number of this problem = 1, we know this problem is well-conditioned.

c) As discussed in class, if algorithm is backward stable, the relative forward error is bounded as follows:

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq \hat{k}^R \cdot \epsilon_{\text{machine}}$$

Because we calculated the value of \hat{k}^R to be equal to 1 in d), we can state that:

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq \epsilon_{\text{machine}}$$

Since the relative forward error is $O(\epsilon_{\text{machine}})$, we can state that the relative forward error is minimal. In other words, we have achieved the "ambitious" error margin.

f) To calculate the eigenvalues, let us first obtain the characteristic polynomial. This can be obtained by:

$$\det(M - \lambda I) = \det \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$= \det \begin{bmatrix} (a-\lambda) & 0 \\ 0 & (a-\lambda) \end{bmatrix}$$

$$\begin{aligned} &= (a-\lambda)^2 \\ &= \lambda^2 - 2a\lambda + a^2 \end{aligned}$$

The steps followed by algorithm V are as follows:

- 1) Obtain the coefficients of the characteristic polynomial
- 2) Equate the characteristic polynomial to zero and calculate its roots.

In step 1 of the algorithm:

- a) Coefficient of $\lambda^2 = 1$. Let us assume that 1 can be stored in the floating point representation exactly.
- b) Coefficient of $\lambda = -2a$. Let us assume that (-2) can be represented in the floating point representation exactly. We already know that a can be represented exactly (without any error). However, errors may happen during multiplication.

$$(-2) \textcircled{*} a = \text{fl}(-2) * \text{fl}(a) * (1 + \epsilon_1) \quad (\text{Floating point arithmetic})$$

$$= -2 * a * (1 + \epsilon_1) \quad (\text{fl}(-2) = -2 \text{ and } \text{fl}(a) = a)$$

Let us call this term as c_1 . $c_1 = -2a(1 + \epsilon_1)$

- c) Constant term = a^2 . Just as before, there is no error in storing a , but error may occur during multiplication.

$$a \textcircled{*} a = \text{fl}(a) * \text{fl}(a) * (1 + \epsilon_2) \quad (\text{Floating point arithmetic error})$$

$$= a^2 * (1 + \epsilon_2) \quad (\text{fl}(a) = a)$$

Let us call this term as c_2 . $c_2 = a^2(1 + \epsilon_2)$

Now, we have a slightly perturbed polynomial to calculate the roots of,

$$\tilde{c}_1 + c_1, \tilde{c}_2 + c_2 = 0$$

Let us use the quadratic formula to calculate the roots:

$$\tilde{c} = \frac{-c_1 \pm \sqrt{c_1^2 - 4c_2}}{2}$$

Let us first calculate the error margin in the discriminant, D . From the actual characteristic polynomial, we know that $D=0$. However, the computer calculates the discriminant using the perturbed coefficients, not of the actual coefficients.

$$D = c_1^2 - 4c_2$$

$$\tilde{D} = \text{fl}(c_1^2) \ominus \text{fl}(4c_2)$$

First, let us calculate $\text{fl}(c_1^2)$ and $\text{fl}(4c_2)$

$$\text{fl}(c_1^2) = \text{fl}(c_1) * \text{fl}(c_1) * (1 + \epsilon_3)$$

$$= (\text{fl}(c_1))^2 (1 + \epsilon_3)$$

$$= (-2a(1 + \epsilon_1))^2 (1 + \epsilon_3)$$

$$= 4a^2 (1 + \epsilon_1)^2 (1 + \epsilon_3)$$

$$= 4a^2 (1 + 2\epsilon_1 + \epsilon_1^2) (1 + \epsilon_3)$$

$$= 4a^2 (1 + 2\epsilon_1) (1 + \epsilon_3)$$

$$(\epsilon_1^2 \ll \epsilon_{\text{machine}})$$

$$fl(c_1^2) = 4a^2(1+2\epsilon_1 + \epsilon_3 + 2\epsilon_1\epsilon_3)$$

$$= 4a^2(1+2\epsilon_1 + \epsilon_3) \quad (2\epsilon_1\epsilon_3 \ll \epsilon_{\text{machine}})$$

$$fl(4c_2) = fl(4) * fl(c_2) * (1+\epsilon_4)$$

$$= 4 * a^2(1+\epsilon_2)(1+\epsilon_4) \quad (\text{Assume } fl(4)=4)$$

$$= 4a^2(1+\epsilon_2 + \epsilon_4 + \epsilon_2\epsilon_4)$$

$$\approx 4a^2(1+\epsilon_2 + \epsilon_4) \quad (\epsilon_2\epsilon_4 \ll \epsilon_{\text{machine}})$$

$$\text{Hence, } \tilde{D} = fl(c_1^2) \oplus fl(4c_2)$$

$$= [fl(c_1^2) - fl(4c_2)](1+\epsilon_5)$$

$$= [4a^2(1+2\epsilon_1 + \epsilon_3) - 4a^2(1+\epsilon_2 + \epsilon_4)](1+\epsilon_5)$$

$$= 4a^2(1+2\epsilon_1 + \epsilon_3 - 1 - \epsilon_2 - \epsilon_4)(1+\epsilon_5)$$

$$= 4a^2(\epsilon_6)(1+\epsilon_5)$$

$$(\text{Assume } (2\epsilon_1 + \epsilon_3 - \epsilon_2 - \epsilon_4) = \epsilon_6)$$

$$= 4a^2(\epsilon_6 + \epsilon_6\epsilon_5)$$

$$= 4a^2\epsilon_6 \quad (\epsilon_5 \ll \epsilon_{\text{mach}})$$

Hence, the discriminant is a small positive number, rather than 0. This causes a drastic change in the roots. If the discriminant is 0, it would result in repeated roots. In this case, because the discriminant is positive, it would result in two distinct roots.

Hence, let us calculate the final roots:

$$\hat{x} = \frac{-c_1 \pm \sqrt{\delta}}{2} = \frac{-(2a(1+\epsilon_1)) \pm \sqrt{4a^2\epsilon_6}}{2}$$

$$= \frac{2a(1+\epsilon_1) \pm \sqrt{4a^2\epsilon_6}}{2}$$

$$(3.3) = \frac{2a(1+\epsilon_1) \pm 2a\sqrt{\epsilon_6}}{2}$$

$$= a(1+\epsilon_1) \pm a\sqrt{\epsilon_6}$$

$$(3.4) \approx a \pm a\sqrt{\epsilon_6} \quad (\text{Because } \sqrt{\epsilon_6} \gg \epsilon_1)$$

g) Let us now calculate forward relative error:

$$\begin{aligned} \text{Forward relative error} &= \frac{|x - x_1|}{|x_1|} = \frac{|a \pm a\sqrt{\epsilon_6} - a|}{|a|} \\ &= \frac{|a\sqrt{\epsilon_6}|}{|a|} \end{aligned}$$

$$= a\sqrt{\epsilon_6}$$

$$= \sqrt{\epsilon_6}$$

Because $\sqrt{\epsilon_6} \gg \epsilon_{\text{machine}}$, the algorithm is not backward stable.

For an algorithm to be backward stable or stable, the forward relative error should be in order of $\epsilon_{\text{machine}}$, i.e. $O(\epsilon_{\text{machine}})$

To further prove that the algorithm is not stable, let us substitute numbers. Assuming a IEEE 754 double precision floating point representation, which has a $\epsilon_{\text{machine}}$ in the order of 10^{-16} .

The relative forward error = $\sqrt{10^{-16}} = 10^{-8}$, which is most definitely not $O(10^{-16})$.