Álgebra Linear Numérica

13/04

Lista 4

Cap. 12

12.1. Suppose A is a 202×202 matrix with $\|A\|_2 = 100$ and $\|A\|_F = 101$. Give the sharpest possible lower bound on the 2-norm condition number $\kappa(A)$.

(12.1) $A \in \mathbb{C}^{202\times 202}$, $\|A\|_2 = 100$, $\|A\|_F = 101$

$\|A\|_2 = \sigma_{max} = 100 \longrightarrow \|A^{-1}\| = \dfrac{1}{\sigma_{min}}$

$\|A\|_F = \sqrt{Tr(A^*A)}$

$\rightarrow Tr(A^*A) = \lambda_1 + \dots + \lambda_{202}$

$10.201 = 10000 + \dots + \lambda_{202}$

$201 = \lambda_2 + \dots + \lambda_{202}$ $\rightarrow$ Logo, para que $K(A) = \|A\| \cdot \|A^{-1}\|$ seja minimizada, $\lambda_{202} = 201$ e o resto dos autovalores, tem de ser 0

$\Rightarrow K(A) \geq \dfrac{100}{\sqrt{201}}$

Cap. 13

13.1. Between an adjacent pair of nonzero IEEE single precision real numbers, how many IEEE double precision numbers are there?

(13.1) IEEE single precison $\overbrace{\epsilon_{machine}}^{\epsilon_s} = 2^{-24} \Rightarrow t = 24$

IEEE double precision $\underbrace{\epsilon_{machine}}_{\epsilon_d} = 2^{-53} \Rightarrow t = 53$

IEEE single precision

$1 \quad \overbrace{\qquad\qquad}^{2^{-23}} \quad 1+2^{-23}$

IEEE double precision

$1 \quad \overbrace{\qquad\qquad}^{2^{-52}} \quad 1+2^{-52}$

Quantas vezes eu multiplico 2^{-52} para chegar em 2^{-23}?

$2^{-52} \cdot \alpha = 2^{-23} \Leftrightarrow \boxed{\alpha = 2^{29}}$

Logo, há $2^{29} - 1$ números representáveis em double precision entre dois representáveis em single precision.

13.2. The floating point system **F** defined by (13.2) includes many integers, but not all of them.

(a) Give an exact formula for the smallest positive integer n that does not belong to **F**.

(b) In particular, what are the values of n for IEEE single and double precision arithmetic?

(c) Figure out a way to verify this result for your own computer. Specifically, design and run a program that produces evidence that $n-3$, $n-2$, and $n-1$ belong to **F** but n does not. What about $n+1$, $n+2$, and $n+3$?

13.2 a) $F = \left\{ \pm \frac{m}{\beta^t} \beta^e \right\}$

Sabemos que $0 \leq m \leq \beta^t$, ou seja, o próximo inteiro $\beta^t + 1$ não é representável no conjunto!

b) $2^{24}+1$

$2^{53}+1$

c) No computador

CAP. 14

14.2. (a) Show that $(1+O(\epsilon_{\text{machine}}))(1+O(\epsilon_{\text{machine}})) = 1+O(\epsilon_{\text{machine}})$. The precise meaning of this statement is that if f is a function satisfying $f(\epsilon_{\text{machine}}) = (1+O(\epsilon_{\text{machine}}))(1+O(\epsilon_{\text{machine}}))$ as $\epsilon_{\text{machine}} \to 0$, then f also satisfies $f(\epsilon_{\text{machine}}) = 1+O(\epsilon_{\text{machine}})$ as $\epsilon_{\text{machine}} \to 0$.

(b) Show that $(1+O(\epsilon_{\text{machine}}))^{-1} = 1+O(\epsilon_{\text{machine}})$.

a) Para facilitar, vou substituir por ϵ_1 e ϵ_2 onde $\epsilon_1 = O(\epsilon_{mac})$, $\epsilon_2 = O(\epsilon_{mac})$ $[\epsilon_1 \leq C_1 \epsilon_{mac}, \epsilon_2 \leq C_2 \epsilon_{mac}]$

$$(1+\epsilon_1)(1+\epsilon_2) = 1 + \epsilon_1 + \epsilon_2 + \epsilon_1 \epsilon_2$$

Provando que $\epsilon_1 \epsilon_2 = O(\epsilon_{mac})$: Consideramos $\epsilon_{mac} \leq 1$

$$|\epsilon_1 \cdot \epsilon_2| \leq C_1 \cdot C_2 \cdot \epsilon_{mac}^2 \Rightarrow |\epsilon_1 \epsilon_2| \leq C \cdot \epsilon_{mac}$$

Provando que $\epsilon_1 + \epsilon_2 = O(\epsilon_{mac})$

$$|\epsilon_1| + |\epsilon_2| \leq C_1 \epsilon_{mac} + C_2 \epsilon_{mac} \Leftrightarrow |\epsilon_1| + |\epsilon_2| \leq \underbrace{(C_1 + C_2)}_{C} \epsilon_{mac}$$

Por desigualdade triangular: $|x+y| \leq |x| + |y|$

$$|\epsilon_1 + \epsilon_2| \leq |\epsilon_1| + |\epsilon_2| \leq C\epsilon_{mac} \Rightarrow \epsilon_1 + \epsilon_2 = O(\epsilon_{mac})$$

Logo: $(1+\epsilon_1)(1+\epsilon_2) = 1 + O(\epsilon_{mac})$

b) $\dfrac{1}{1+O(\epsilon_{mac})} \rightarrow \dfrac{1}{1+\epsilon} = \underbrace{1-\epsilon+\epsilon^2-\epsilon^3+\dots}_{O(\epsilon_{mac})}$

ou seja $\dfrac{1}{1+O(\epsilon_{mac})} = 1 + O(\epsilon_{mac})$

CAP. 15

15.1. Each of the following problems describes an algorithm implemented on a computer satisfying the axioms (13.5) and (13.7). For each one, state whether the algorithm is *backward stable*, *stable but not backward stable*, or *unstable*, and prove it or at least give a reasonably convincing argument. Be sure to follow the definitions as given in the text.

(a) Data: $x \in \mathbb{C}$. Solution: $2x$, computed as $x \oplus x$.

$f(x) = x + x$ $\qquad \tilde{f}(x) = 2x(1+\epsilon)$

$\tilde{f}(x) = x \oplus x$ $\qquad$ defina $\tilde{x} = x(1+\epsilon)$ → Temos $f(\tilde{x}) = \tilde{f}(x)$

→ $f(\tilde{x}) = x(1+\epsilon) + x(1+\epsilon) = 2x(1+\epsilon)$

→ Backward stable

$\dfrac{|x(1+\epsilon)-x|}{|x|} = \dfrac{\cancel{|x|}\,|\epsilon|}{\cancel{|x|}} = |\epsilon| = O(\epsilon_{mac})$

(b) Data: $x \in \mathbb{C}$. Solution: x^2, computed as $x \otimes x$.

$f(x) = x \cdot x$

$\tilde{f}(x) = x \odot x$

$\dfrac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = \dfrac{\|x^2(1+\epsilon) - x^2\|}{\|x^2\|} = |\epsilon| = O(\epsilon_{mac})$

Estável!

$\tilde{f}(x) = x^2 \cdot (1+\epsilon)$ $\qquad f(\tilde{x}) = x\sqrt{1+\epsilon} \cdot x\sqrt{1+\epsilon}$

defina $\tilde{x} = x\sqrt{(1+\epsilon)}$ $\qquad = x^2 \cdot (1+\epsilon) = \tilde{f}(x)$

$\dfrac{|\tilde{x}-x|}{|x|} = \dfrac{|x\sqrt{1+\epsilon} - x|}{|x|} = \dfrac{|x|\,|\sqrt{1+\epsilon}-1|}{|x|} = |\sqrt{1+\epsilon}-1|$

$\sqrt{1+\epsilon} = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3 - \dots$

$\Rightarrow |\frac{1}{2}\epsilon - \frac{1}{8}\epsilon^2 + \frac{1}{16}\epsilon^3 - \dots| = O(\epsilon_{mac})$

Provei no exercício 14.2 que $\epsilon_i \cdot \epsilon_j$ com $\epsilon_i = O(\epsilon_{mac})$ e $\epsilon_j = O(\epsilon_{mac})$ é $O(\epsilon_{mac})$ e $\epsilon_i + \epsilon_j$ também é ϵ_{mac}.

(c) Data: $x \in \mathbb{C} \setminus \{0\}$. Solution: 1, computed as $x \oplus x$. (A machine satisfying (13.6) will give exactly the right answer, but our definitions are based on the weaker condition (13.7).)

$$f(x) = x \div x$$
$$\tilde{f}(x) = x \oslash x$$

$$\tilde{f}(x) = 1(1+\epsilon_{mac})$$

Defina $\tilde{x} = 1 + \epsilon_{mac}$

$$f(\tilde{x}) = \frac{\tilde{x}}{\tilde{x}}$$ → Não pode ser Backward stable (Nunca da $1+\epsilon$)

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = \frac{|1+\epsilon-1|}{|1|} = |\epsilon| = O(\epsilon_{mac})$$

Estável!

(d) Data: $x \in \mathbb{C}$. Solution: 0, computed as $x \ominus x$. (Again, a real machine may do better than our definitions based on (13.7).)

$$f(x) = x - x$$
$$\tilde{f}(x) = x \ominus x$$

$$\tilde{f}(x) = (x-x)(1+\epsilon)$$
$$\tilde{f}(x) = 0$$

Seja $\tilde{x} = x(1+\epsilon)$, com $\epsilon = O(\epsilon_{mac})$

$$f(\tilde{x}) = \tilde{x} - \tilde{x} = 0$$

$$\|\tilde{f}(x) - f(x)\| \leq C \cdot \epsilon_{mac} \cdot \|f(x)\|$$

$$0 \leq C \cdot \epsilon_{mac} \cdot 0 \Rightarrow \checkmark \text{ Estável}$$

$$\frac{\|\tilde{x} - x\|}{\|x\|} = \epsilon = O(\epsilon_{mac})$$

É backward's stable!

(e) Data: none. Solution: e, computed by summing $\sum_{k=0}^{\infty} 1/k!$ from left to right using $\otimes$ and $\oplus$, stopping when a summand is reached of magnitude $< \epsilon_{\text{machine}}$.

$$e = \sum_{k=0}^{\infty} \frac{1}{f(k)} \quad \text{onde} \quad f(k) = k \otimes (k-1) \otimes \ldots \otimes 1$$

$f(k)$ é estável? (k é inteiro) Provamos que SIM, f é estável e backward's stable

$$e = 1 \oplus (1 \oslash f(2)) \oplus (1 \oslash f(3)) \oplus \dots$$

Como $\oslash$, $f(k)$ e $\oplus$ são backward stable, como mostrei antes, calcular e é backward stable

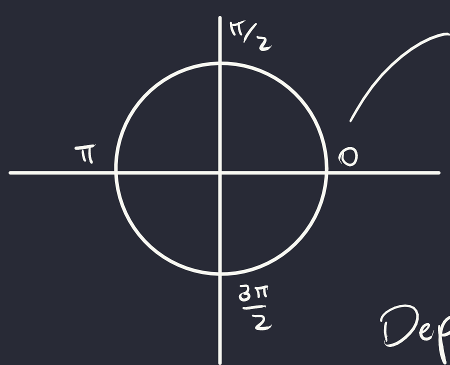
(f) Data: none. Solution: e, computed by the same algorithm as above except with the series summed from right to left.

$$F(k) = \sum_{j=1}^{k} \frac{1}{f(j)} \qquad f(x) = x \otimes (x-1) \otimes (x-2) \otimes \dots \otimes 1$$

$$F(k) = \frac{1}{f(k)} \oplus \frac{1}{f(k-1)} \oplus \frac{1}{f(k-2)} \oplus \dots \oplus \frac{1}{f(2)} \oplus 1$$

A função é estável, mas não é backward stable se eu coloco k muito alto, muitos erros vão se acumular de forma que eu não vou ter um $\frac{\|\tilde{x} - x\|}{\|x\|} = O(\epsilon)$

(g) Data: none. Solution: π, computed by doing an exhaustive search to find the smallest floating point number x in the interval $[3, 4]$ such that $s(x) \otimes s(x') \leq 0$. Here $s(x)$ is an algorithm that calculates $\sin(x)$ stably in the given interval, and x' denotes the next floating point number after x in the floating point system.



$\text{sen}(x) = x(1+\epsilon)$

↳ operação stable

$\otimes$ ↳ operação stable

Depende apenas da escolha dos números na hora da checagem de condição, logo, é backwards stable