1.9 condition and stability of functions

condition of a function $f(x)$

The Problem:

Given a function

$$f : \mathbb{R}^m \to \mathbb{R}^k$$

compute the function value $f(x)$ for some $x \in \mathbb{R}^m$

Definition:

The *(relative) condition number* of a function is

$$\kappa(x) = \sup_{y \neq x} \frac{\|f(y) - f(x)\|/\|f(x)\|}{\|y - x\|/|x\|}$$

a local version is

$$\kappa(x) = \lim_{\epsilon \to 0} \sup_{\|y-x\|<\epsilon} \frac{\|f(y) - f(x)\|/\|f(x)\|}{\|y - x\|/|x\|}$$

or simplified $y = (1 + \epsilon S)x$ where S is a diagonal matrix with ± 1 diagonal elements

$$\kappa(x) = \lim \sup \frac{\|f((1 + \epsilon S)x) - f(x)\|}{\ldots}$$

examples

1. $f(x) = 10x + 5$ (both global and local version are the same)

$$\kappa(x) = \sup_y \frac{10(x-y)/(10x+5)}{(x-y)/x}$$
$$= \frac{10x}{10x+5}$$

2. $f(x) = \sqrt{x}$ for $x > 0$

$$\kappa(f) = \sup_{y>0} \frac{(\sqrt{x}-\sqrt{y})/\sqrt{x}}{(x-y)/x}$$
$$= \sup_{y>0} \frac{\sqrt{x}}{\sqrt{x}+\sqrt{y}} = 1$$

- the local version is $\kappa(f) = 0.5$

the difference $f(x_1, x_2) = x_1 - x_2$ can be ill-conditioned

$$\kappa(x) = \sup \frac{|x_1 - x_2 - y_1 + y_2| / |x_1 - x_2|}{\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \Big/ \sqrt{x_1^2 + x_2^2}}$$

- maximum obtained for $x_1 - y_1 = -(x_2 - y_2)$ and thus

$$\kappa(x) = \sqrt{2 \frac{x_1^2 + x_2^2}{(x_1 - x_2)^2}}$$

- condition number large for $x_1 \approx x_2$

the exponential function

- $f(x) = \exp(x)$ for $x \in [0, M]$

$$\kappa(x) = \sup_{0 \le y \le M} \frac{|e^y - e^x|/e^x}{|y-x|/|x|} = \sup_y \frac{e^{y-x}-1}{|y-x|}|x| < e^M|x|$$

  - as $|y - x| \le M$ and

$$\frac{e^{y-x}-1}{y-x} = e^{\theta(y-x)}$$

    for some $\theta \in [0,1]$ because the left hand side is the slope of a secant ...

- the local condition number is $\kappa(f) = |x|$

condition number of a matrix

- matrix-vector product $f(x) = Ax$ for $x \in \mathbb{R}^n$

$$\begin{aligned}\kappa(A) &= \sup \frac{\|A(x-y)\|/\|Ax\|}{\|x-y\|/\|x\|} \\ &= \sup \frac{\|A(x-y)\|}{\|x-y\|} \cdot \frac{\|x\|}{\|Ax\|} = \|A\| \cdot \|A^{-1}\|\end{aligned}$$

- it follows that $\kappa(A) = \kappa(A^{-1})$

stability of numerical function $f(x, \delta)$

$$f : \mathbb{R}^m \otimes \mathbb{R}^k \to \mathbb{R}$$

models a function as evaluated on a computer

- where $\delta \in \mathbb{R}^k$ is an error parameter
- $f(x, 0)$ is the exact value

Definition (stability)

$f(x, \delta)$ is *stable* if for any choice of

- $x \in \mathbb{R}^m$
- $\epsilon > 0$ and $\delta \in \mathbb{R}^k$ with $|\delta_k| \leq \epsilon$

there exist

- $y \in \mathbb{R}^m$ and $C_1, C_2 > 0$

such that x is close to y, i.e.,

a stronger and simpler condition

- concept used mostly in actual analysis

Definition (backward stability)

$f(x, \delta)$ is *backward stable* if for any choice of

- $x \in \mathbb{R}^m$
- $\epsilon > 0$ and $\delta \in \mathbb{R}^k$ with $|\delta_k| \leq \epsilon$

there exist

- $y \in \mathbb{R}^m$
- $C > 0$

such that x is close to y, i.e.,

$$\frac{\|y - x\|}{\|x\|} \leq C\epsilon$$

and $f(x, \delta)$ is equal to $f(y, 0)$

accuracy of a backward stable algorithm

Definition: relative error

$$e = \frac{f(x,\delta) - f(x,0)}{|f(x,0)|}$$

Proposition

If $f(x,\delta)$ is backward stable and $f(x,0)$ is well conditioned with condition number $\kappa(x)$, then there is a $C > 0$ such that the relative error satisfies

$$|e| \leq \kappa(x)\, C\epsilon$$

for all rounding errors δ with $|\delta_k| \leq \epsilon$

Proof.

by backward stability and the definition of the condition number one has from backward stability some y such that

$$
\begin{aligned}
\frac{|f(x,\delta) - f(x,0)|}{|f(x,0)|} &= \frac{|f(y,0) - f(x,0)|}{|f(x,0)|} \\
&\leq \kappa(x)\,\frac{\|y - x\|}{\|x\|} \\
&\leq C\kappa(x)\,\epsilon
\end{aligned}
$$

where $\|y - x\|/\|x\| \leq C\epsilon$

■

Remarks

- The constant C depends on the algorithm and in particular the dimension of δ
- Often it is easier to determine the constant C and κ then bounding the error directly
- When applied to the difference one sees that the

example: $a - bc/d$ (Schur complement)

$$
\begin{aligned}
u_1 &= a \\
u_2 &= b \\
u_3 &= c \\
u_4 &= d \\
u_5 &= u_2 u_3 \\
u_6 &= u_5 / u_4 \\
u_7 &= u_1 - u_6
\end{aligned}
$$

- input $x = (a, b, c, d)$ (components of 2 by 2 matrix)
- Schur complement is major tool for Gaussian elimination
- backward stability has been used to get rounding error bounds for Gaussian elimination to differentiate between the effects of the algorithm and the effects of the data (the matrix)

example: $a - bc/d$ with rounding errors

$$
\begin{aligned}
v_1 &= (1+\delta_1)\, a \\
v_2 &= (1+\delta_2)\, b \\
v_3 &= (1+\delta_3)\, c \\
v_4 &= (1+\delta_4)\, d \\
v_5 &= (1+\delta_5)\, v_2 v_3 \\
v_6 &= (1+\delta_6)\, v_5 / v_4 \\
v_7 &= (1+\delta_7)\, (v_1 - v_6)
\end{aligned}
$$

example: $a - bc/d$ backward stable model

$$
\begin{aligned}
z_1 &= (1 + \eta_1)\, a \\
z_2 &= (1 + \eta_2)\, b \\
z_3 &= (1 + \eta_3)\, c \\
z_4 &= (1 + \eta_4)\, d \\
z_5 &= z_2 z_3 \\
z_6 &= z_5 / z_4 \\
z_7 &= z_1 - z_6
\end{aligned}
$$

- the η_k are a function of the δ_j
- the result is the same as before $z_7 = v_7$

example: $a - bc/d$ – compute the η_j

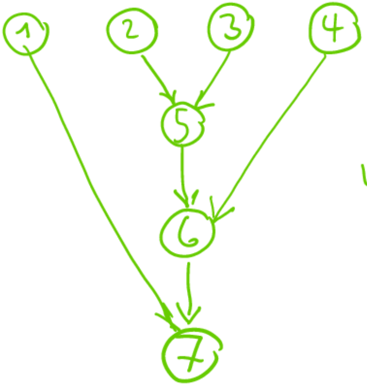
$$
\begin{aligned}
z_7 &= v_7 = (1+\delta_7)(v_1 - v_6) = z_1 - z_6 \\
z_6 &= (1+\delta_7)v_6 = (1+\delta_7)(1+\delta_6)v_5/v_4 = z_5/z_4 \\
z_5 &= (1+\delta_7)v_5 = (1+\delta_7)(1+\delta_5)v_2v_3 = z_2z_3 \\
z_4 &= (1+\delta_6)^{-1}v_4 = (1+\delta_6)^{-1}(1+\delta_4)d = (1+\eta_4)d \\
z_3 &= (1+\delta_7)v_3 = (1+\delta_7)(1+\delta_3)c = (1+\eta_3)c \\
z_2 &= (1+\delta_5)v_2 = (1+\delta_5)(1+\delta_2)b = (1+\eta_2)b \\
z_1 &= (1+\delta_7)v_1 = (1+\delta_7)(1+\delta_1)a = (1+\eta_1)a
\end{aligned}
$$

- thus one gets for the η_j

$$
\begin{aligned}
\eta_1 &= (1+\delta_7)(1+\delta_1) - 1 \\
\eta_2 &= (1+\delta_5)(1+\delta_2) - 1 \\
\eta_3 &= (1+\delta_7)(1+\delta_3) - 1 \\
\eta_4 &= (1+\delta_6)^{-1}(1+\delta_4) - 1
\end{aligned}
$$

Math3511 - graph of Schur complement

Wednesday, 7 March 2018 10:33 AM



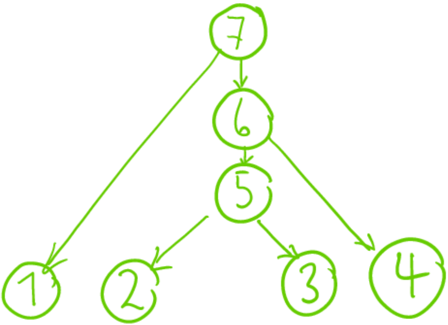
inputs $\underline{x}$

$y = f(\underline{x})$

output y

Math3511 - inverse graph

Wednesday, 7 March 2018 10:36 AM



another example $f(x) = 1 + x$

- usual (global) error analysis from section 1.8

$$
\begin{aligned}
v_1 &= (1+\delta_1)x \\
v_2 &= (1+\delta_2)(1+v_1)
\end{aligned}
$$

- this gives for the result $v_2 = f(x, \delta)$ with $\delta = (\delta_1, \delta_2)$

  $$v_2 = (1+\delta_2)(1+(1+\delta_1)x) = (1+\theta_2)(1+x)$$

  and from this one gets (neglecting small terms like $\delta_1\delta_2$ for the relative error θ_2

  $$
  \begin{aligned}
  \theta_2 &= \frac{(1+\delta_2)(1+(1+\delta_1)x)}{1+x} - 1 \\
  &\approx \frac{x}{1+x}\delta_1 + \delta_2
  \end{aligned}
  $$

  thus the relative error is bounded by $(C+1)\epsilon$ if $|x|/|1+x| \leq C$ which we take as domain of f

backward stability of $f(x, \delta)$ from previous slide

- $f(x, \delta)$ is backward stable if there is a ζ_1 such that

$$f(x, \delta) = v_2 = z_2$$

for some z_1, z_2 and ζ_1 with

$$z_1 = (1 + \zeta_1)x$$
$$z_2 = 1 + z_1$$

- solving backwards gives

$$1 + z_1 = z_2 = v_2 = 1 + (1 + \delta_2)v_1 + \delta_2$$

- and so

$$z_1 = (1 + \delta_2)v_1 + \delta_2 = (1 + \delta_2)(1 + \delta_1)x + \delta_2 = \zeta_1 x$$

and consequently

$$\zeta_1 = (1 + \delta_2)(1 + \delta_1) + \delta_2/x$$

- thus our "algorithm" $f(x, \delta)$ is backward stable if $|x| > 1/M > 0$

condition number of $f(x) = 1 + x$

- the condition number of f is

$$\kappa(f) = \sup_y \frac{|f(y) - f(x)|}{|y - x|} \frac{|x|}{|f(x)|}$$
$$= \frac{|x|}{|1 + x|}$$

- the condition number is large if $x \approx -1$ where the function is ill-conditioned but the function is well-conditioned otherwise