Lecture 19: Error Analysis

The last few lectures have studied stability: the sensitivity of a mathematical problem to perturbations in the input. We now turn our attention to the actual error in numerical algorithms. We model this as follows: given a problem $f : Y \to Z$ between two normed vector spaces, consider an algorithm as another problem $\tilde{f} : Y \to Z$ that we hope satisfies $f(x) \approx \tilde{f}(x)$.

There are two types of errors we can look at: *forward* and *backward* error. If these errors are "small" we say that the algorithm is *forward stable* or *backward stable*. The definition of "small" depends on the context.

Forward error

The *absolute forward error* is defined as

$$\|\tilde{f}(x) - f(x)\|_Z$$

while the *relative forward error* is

$$\frac{\|\tilde{f}(x) - f(x)\|_Z}{\|f(x)\|_Z}$$

Backward error

Suppose that there exists a Δx so that $\tilde{f}(x) = f(x + \Delta x)$. Then the *absolute backward error* is defined as

$$\|\Delta x\|_Y$$

and the *relative backward error* isdefined as

$$\frac{\|\Delta x\|_Y}{\|x\|_Y.}$$

Warning: Backward error may not always be defined: for example, if $f(x) = 1$ and $\tilde{f}(x) = 0$ we have $f(x + \Delta x) = 1 \neq 0$ for all Δx. In this case, we can define the backward error as ∞.

Floating point error analysis

We now consider the forward and backward error in algorithms arising from floating point arithmetic.

Here $\mathrm{fl}(x)$ denotes the operator of rounding a number to floating point. This is always exact up to the last bit in the significand. If there are p bits used to represent the significand, define *machine epsilon* as

$$\epsilon_m \triangleq 2^{-p}.$$

Since we round to nearest bit, we have the property that

$$\mathrm{fl}(x) = x(1+\delta_x)$$

where $|\delta_x|| \leq \frac{\epsilon_m}{2}$.

We can verify this numerically. `eps(Float32)` and `eps(Float64)` give machine epsilon for `Float32` and `Float64` respectively. For `Float32` we have $p = 23$, so this means it returns 2^{-23}:

In [24]:

```
eps(Float32),2.0f0^(-23)
```

Out[24]:

```
(1.1920929f-7,1.1920929f-7)
```

We have $\mathrm{fl}(x) = x(1+\delta_x)$, which implies that

$$|x - \mathrm{fl}(x)| = |x||\delta_x| \leq |x|\frac{\epsilon_m}{2}.$$

We confirm this for a simple example:

In [13]:

```
x=1/3
x̃=Float32(x)

abs(x-x̃) ≤ abs(x)*εm/2
```

Out[13]:

```
true
```

We can explain this using the case $x = 1$. If we add $\epsilon_m/2$ to one we round back to one:

In [25]:

```
Float32(1.0+2.0^(-24))
```

Out[25]:

```
1.0f0
```

Any perturbation above this, no matter how small, rounds up:

In [26]:

```
Float32(1.0+2.0^(-24)+2.0^(-30))
```

Out[26]:

```
1.0000001f0
```

Thus the worst case is off by the last bit, but within half:

In [27]:

```
bits(Float32(1.0+2.0^(-24)+2.0^(-25)))
```

Out[27]:

```
"00111111100000000000000000000001"
```

In [28]:

```
bits(Float32(1.0))
```

Out[28]:

```
"00111111100000000000000000000000"
```

Example 1: error analysis for rounding a number

We now consider the backward and forward error of rounding. Define $f(x) = x$ and $\tilde{f}(x) = \mathrm{fl}(x)$, where $f, \tilde{f} : \mathbb{R} \to \mathbb{R}$ with the absolute value norm attached, which denote as $|\cdot| \to |\cdot|$.

Forward error:

$$\frac{|f(x) - \tilde{f}(x)|}{|f(x)|} = \frac{|x - x(1+\delta_x)|}{|x|} = |\delta_x| \leq \frac{\epsilon_m}{2}.$$

Backward error: Since $\tilde{f}(x) = x(1+\delta_x) = f(x + x\delta_x) = f(x + \Delta x)$ for $\Delta x = x\delta_x$, we have the (relative) backward error

$$\frac{|\Delta x|}{|x|} = |\delta_x| \leq \frac{\epsilon_m}{2}$$

Example 2: error analysis for adding two floating point numbers

Assume x and y are floating point numbers. Consider the problem $f(x, y) = x + y$ calculated via the algorithm $\tilde{f}(x, y) = x \oplus y = \mathrm{fl}(x + y) = (x + y)(1 + \delta_{x+y})$ where $f, \tilde{f} : \mathbb{R}^2 \to \mathbb{R}$ with norms $\|\cdot\|_\infty \to |\cdot|$.

Forward error:

$$\frac{|f(x) - \tilde{f}(x)|}{|f(x)|} = \frac{|x + y - (x+y)(1+\delta_{x+y})|}{|x+y|} = |\delta_{x+y}| \leq \frac{\epsilon_m}{2}.$$

Backward error: Since $\tilde{f}(x, y) = f(x + x\delta_{x+y}x, y + y\delta_{x+y}) = f(x + \Delta x, y + \Delta y)$, we have the backward error

$$\frac{\left\| \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right\|_\infty}{\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_\infty} = |\delta_{x+y}| \leq \frac{\epsilon_m}{2}.$$

Example 3: error analysis for adding two real numbers

Assume x and y are general real numbers. Again consider the problem $f(x, y) = x + y$, but now calculated via the algorithm

$$\tilde{f}(x, y) = \mathrm{fl}(x) \oplus \mathrm{fl}(y) = x(1 + \delta_x) \oplus y(1 + \delta_y) = (x(1 + \delta_x) + y(1 + \delta_y))(1 + \delta_z) = x + y + x(\delta_x$$

where $z = x(1 + \delta_x) + y(1 + \delta_y)$. As before, $f, \tilde{f} : \mathbb{R}^2 \to \mathbb{R}$ with norms $\| \cdot \|_\infty \to | \cdot |$.

Backward error: Since $\tilde{f}(x, y) = f(x + x\delta_{x+y}x, y + y\delta_{x+y}) = f(x + \Delta x, y + \Delta y)$, we have the backward error

$$\frac{\left\| \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right\|_\infty}{\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_\infty} = |\delta_{x+y}| \leq \frac{\epsilon_m}{2}.$$

Forward error:

$$\frac{|f(x) - \tilde{f}(x)|}{|f(x)|} = \frac{|x + y - (x + y)(1 + \delta_{x+y})|}{|x + y|} = |\delta_{x+y}| \leq \frac{\epsilon_m}{2}.$$