Set 4: Basics Part 2

Kyle A. Gallivan

Department of Mathematics

Florida State University

Foundations of Computational Math 1
Fall 2012

Finite Precision

- All discussions so far have assumed we can compute with elements of $\mathbb{R}$ and by extension $\mathbb{C}$.
- Representation of numbers was symbolic and therefore exact, e.g., π, e^x, $1/3$
- arithmetic was symbolic and therefore exact.
- Computers store information using a finite number of digits (bits).
- Computers do not perform exact arithmetic.

Finite Precision

Consequently,

- The problem solved is not necessarily the problem posed.
- The computed solution does not necessarily solve the original problem or the problem stored in the computer.
- The computed solution may solve a problem that is not representable in the computer.

Numerical Analysis and Finite Precision

- Numerical error analysis addresses the effects of finite precision of representation and computation.
- Topics:
  - Representation
  - Arithmetic
  - Errors: forms and bounds
  - Conditioning of problem
  - Stability of an algorithm

Sources

In addtion to the textbook the following are sources for this presentaion and useful references (see also their citations):

- *What Every Computer Scientist Should Know About Floating-Point Arithmetic* David Goldberg, ACM Computing Surveys, March, 1991. `http://docs.sun.com/source/806-3568/ncg_goldberg.html`
- *Matrix Algorithms, Volume 1: Basic Decompositions*, G. W. Stewart, SIAM, 1998.
- *Accuracy and Stability of Numerical Algorithms*, N. J. Higham, SIAM, Second Edition, 2002.
- *The Handbook of Mathematical Functions*, M. Abramowitz and I. Stegun, Tenth Printing, available on the web at several URLs

Integers – finite number exactly represented

Binary	unsigned	sign magnitude	excess three	two's complement
000	0	0	-3	0
001	1	1	-2	1
010	2	2	-1	2
011	3	3	0	3
100	4	-0	1	-4
101	5	-1	2	-3
110	6	-2	3	-2
111	7	-3	4	-1

Real Number Representation

Real numbers can be written as decimal expansions:

$$\pi = 3.14159265358979\ldots$$

$$\frac{1}{3} = 0.3333333\ldots$$

$$98.6 = 986 \times 10^{-1} = 0.986 \times 10^{2} = (\frac{9}{10} + \frac{8}{100} + \frac{6}{1000}) \times 10^{2}$$

$$\forall x \in \mathbb{R} \quad x = \pm 10^{e} \sum_{k=1}^{\infty} d_k \times 10^{-k}, \;\; 0 \leq d_k \leq 9, \quad d_1 \neq 0, \quad e \in \mathbb{Z}$$

Floating Point Representation

A floating point number system is characterized by four integers

- β the base or radix
- t the precision
- exponent bounds $L \leq e \leq U$

Definition 4.1. A floating point number system $F \subset \mathbb{R}$ is a set of real numbers of the form

$$y = \pm m \times \beta^{e-t}$$

$$y = \pm \beta^e \sum_{k=1}^{t} d_k \times \beta^{-k} = \pm \beta^e (0.d_1 d_2 \ldots d_t)$$

where $m \in \mathbb{Z}$, $0 \leq m \leq \beta^t - 1$.

Floating Point Representation

Definition 4.2. A floating point number system $\mathcal{F} \subset \mathbb{R}$ is normalized if the mantissa, m, (or significand or fraction) satisfies $\beta^{t-1} \leq m < \beta^t$ when $y \neq 0$. This is the same as $d_1 \neq 0$. It is assumed that 0 has a special representation.

$$-9.9 = -0.99 \times 10^1$$

$$22 = +.1011 * 2^5$$

$$\frac{3}{8} = +.1200 \times 4^0$$

Example

The system with $\beta = 2$, $t = 3$, $L = -1$, $U = 3$ has the following normalized positive numbers. Note the exponents are written as a base 10 integer for clarity.

$$.100 \times 2^{-1} = 0.25, \quad .101 \times 2^{-1} = 0.3125,$$

$$.110 \times 2^{-1} = 0.375, \quad .111 \times 2^{-1} = 0.4375$$

$$.100 \times 2^{0} = 0.5, \quad .101 \times 2^{0} = 0.625, \quad .110 \times 2^{0} = 0.75, \quad .111 \times 2^{0} = 0.875$$

$$.100 \times 2^{1} = 1.0, \quad .101 \times 2^{1} = 1.25, \quad .110 \times 2^{1} = 1.5, \quad .111 \times 2^{1} = 1.75$$

$$.100 \times 2^{2} = 2.0, \quad .101 \times 2^{2} = 2.5, \quad .110 \times 2^{2} = 3.0, \quad .111 \times 2^{2} = 3.5$$

$$.100 \times 2^{3} = 4.0, \quad .101 \times 2^{3} = 5.0, \quad .110 \times 2^{3} = 6.0, \quad .111 \times 2^{3} = 7.0$$

Floating Point Representation

- $\mathcal{F}$ is a finite set with $1 + 2(U - L + 1)(\beta - 1)\beta^{t-1}$ elements
  - $(\beta - 1)\beta^{t-1}$ normalized mantissas
  - $(U - L + 1)$ exponents
  - $\pm \rightarrow 2$ values for each mantissa, exponent pair
  - 0 represented via special pattern
- 41 for $\beta = 2$, $t = 3$, $L = -1$, $U = 3$ example.
- $\beta^{L-1} \leq |y| \leq \beta^{U}(1 - \beta^{-t})$
- absolute difference between elements is β^{e-t}
- machine epsilon is $\epsilon_M = \beta^{1-t}$. (distance from 1 to next $x \in \mathcal{F}$)
- magnitude of the relative distance varies periodically: $1/m(x)$.

Example

Definition 4.3. Subnormal or denormalized elements are those with $e = L$, value $\pm m \times \beta^{L-t}$ and $d_1 = 0$. The smallest denormalized number is

$$\mu = \beta^{L-t} = \beta^{L-1}\beta^{1-t} = y_{min}\epsilon_M$$

The system with $\beta = 2$, $t = 3$, $L = -1$, $U = 3$ has the following denormalized numbers

$$.001 \times 2^{-1} = \frac{1}{16} = 0.0625$$
$$.010 \times 2^{-1} = \frac{1}{8} = 0.125$$
$$.011 \times 2^{-1} = \frac{3}{16} = 0.18755$$

These are included in the system to facilitate what is known as gradual underflow.

Mapping Real to Floating Point

- Every floating point number is a real number but not vice versa.
- We need a map $fl(x) : \mathbb{R} \to \mathcal{F}$.
- Care must be taken since the range of $\mathcal{F}$ is limited.
- Many possible versions of the map $fl(x)$ but not all are robust.

Mapping Real to Floating Point

Suppose we have two consecutive positive normalized floating point numbers $f_1 < f_2$ and a real number $f_1 < x < f_2$.

$$x = (0.d_1 \ldots d_{t-1} d_t d_{t+1} d_{t+2} \ldots) \times \beta^e$$

$$f_1 = (0.d_1 \ldots d_{t-1} d_t) \times \beta^e$$

$$f_2 = (0.d_1 \ldots d_{t-1} \tilde{d}_t) \times \beta^e \text{ where } \tilde{d}_t = d_t + 1$$

$$fl(x) = f_1 \text{ called chopping}$$

$$\frac{x - fl(x)}{x} = \frac{(0.0 \ldots 00 d_{t+1} d_{t+2} \ldots) \times \beta^e}{(0.d_1 \ldots d_{t-1} d_t d_{t+1} d_{t+2} \ldots) \times \beta^e} \le \frac{\beta^{-t}}{\beta^{-1}} = \beta^{1-t} = \epsilon_M$$

Mapping Real to Floating Point

If rather than chopping we have rounding then

$$fl(x) = \begin{cases} f_1 & \text{when } |x - f_1| < |x - f_2| \\ f_2 & \text{when } |x - f_2| < |x - f_1| \end{cases}$$

Ties can be broken in different ways.

- round to even, i.e., d_t is even in $fl(x)$
- round to odd, i.e., d_t is odd in $fl(x)$
- round towards 0
- round towards ∞

Mapping Real to Floating Point

Lemma 4.1. *For any x in the range of the floating point system $\mathcal{F}(\beta, t, L, U)$ we have*

$$\frac{|fl(x) - x|}{|x|} = \begin{cases} \beta^{1-t} = u & \textit{for chopping} \\ \frac{1}{2}\beta^{1-t} = u & \textit{for rounding} \end{cases}$$

$$fl(x) = x(1+\delta), \quad |\delta| < u$$

If x is outside the range of $\mathcal{F}$ then $fl(x)$ is said to overflow.

If $0 \leq |x| \leq y_{min} = \beta^{L-1}$ then then $fl(x)$ is said to underflow.

u is called unit roundoff.

Mapping Real to Floating Point

In practice, x will be a real number produced on the computer that has the form

$$x = (0.d_1 \ldots d_{t-1} d_t d_{t+1} d_{t+2} \ldots d_{t+g}) \times \beta^e$$

That is, it will have some extra digits called guard digits (and sometimes an extra sticky digit).

In this case rounding is based on the value of one or all of the digits $d_{t+1} d_{t+2} \ldots d_{t+g}$, i.e., $x - f_1$ and $x - f_2$.

For example, rounding with a tie rounded up in magnitude is often described as

$$fl(x) = \begin{cases} f_1 & \text{if } d_{t+1} < \frac{\beta}{2} \\ f_2 & \text{if } d_{t+1} \geq \frac{\beta}{2} \end{cases}$$

Mapping Real to Floating Point

Let $\beta = 10$ and $t = 5$

$$x = 142.52731$$

$$y = 142.52500$$

$$fl(x) = \begin{cases} 142.52 & \text{chopping} \\ 142.53 & \text{rounding} \end{cases}$$

$$fl(y) = \begin{cases} 142.52 & \text{chopping} \\ 142.52 & \text{rounding to even} \end{cases}$$

IEEE Format

Single Precision:

- 32-bit FP number
- 1 bit, σ, for the sign of the mantissa
- 8-bit biased exponent, ϵ
- 23(+1)-bit normalized mantissa, μ.
- unit roundoff $u = 2^{-24} \approx 5.96 \times 10^{-8}$

The normalization is a leading hidden bit of 1 that with coefficient 2^0 that is not stored explicitly.

IEEE Format

Double Precision:

- 64-bit FP number
- 1 bit, σ, for the sign of the mantissa
- 11-bit biased exponent, ϵ
- 52(+1)-bit normalized mantissa, μ.
- unit roundoff $u = 2^{-53} \approx 1.11 \times 10^{-16}$

The normalization is a leading hidden bit of 1 that with coefficient 2^0 that is not stored explicitly.

Decoding

If σ, ϵ and μ are the decimal integer values of the bit patterns in the sign, exponent and mantissa fields respectively, then the number represented has the values:

Single precision: ($\epsilon = 0$ and $\epsilon = 255$ have special meanings.)

$$\begin{array}{rcl} sp & = & (-1)^{\sigma} \quad 1.\,\mu \times 2^{\epsilon-127} \\ 1 \leq & \epsilon & \leq 254 \\ 10^{-38} \leq & sp & \leq 10^{38} \quad roughly \end{array}$$

Double precision: ($\epsilon = 0$ and $\epsilon = 2047$ have special meanings.)

$$\begin{array}{rcl} dp & = & (-1)^{\sigma} \quad 1.\,\mu \times 2^{\epsilon-1023} \\ 1 \leq & \epsilon & \leq 2046 \\ 10^{-308} \leq & dp & \leq 10^{308} \quad roughly \end{array}$$

Single Precision Example

32-bit word has fields:

$$\left[\begin{array}{c|c|c} \sigma & \epsilon & \mu \end{array}\right]$$

If $x = -1.5$ we have

$$x = -1.1 \times 2^0$$

$$\sigma = 1$$

$$\epsilon = 0 + 127 = 01111111$$

$$\mu = 10000000000000000000000$$

$$\left[\begin{array}{c|c|c} 1 & 01111111 & 10000000000000000000000 \end{array}\right]$$

Conditioning

Suppose we are to solve a problem specified by data d to yield a solution s. (d and s can be a collection of real scalars.)

The problem can be mathematically represented by

$$F(s, d) = 0 \text{ or } s = f(d)$$

Note this not an algorithm or a program. It is the mathematical specification of the mapping from the data d to the solution s.

Conditioning

Function evaluation:

$$d = \alpha \in \mathbb{R}, \quad s = f(d) \to y = e^{\alpha}$$

Root finding:

$$d = \begin{pmatrix} \alpha & \beta & \gamma \end{pmatrix} \; s = \begin{pmatrix} x_+ & x_- \end{pmatrix} \; \to F(d,s) = 0 = \alpha x^2 + \beta x + \gamma$$

Solving linear systems:

$$d = \left(A \in \mathbb{R}^{n \times n}, \; b \in \mathbb{R}^n\right)$$

$$s = x \to F(s,d) = b - Ax \text{ or } s = x = f(d) = A^{-1}b$$

Conditioning

- $s = f(d)$ desired.
- $fl(d) = d(1+\delta)$ is stored.
- $\tilde{s} = f(fl(d))$ is exact solution to the perturbed problem.
- How much does the perturbation change the solution?

$$e_{abs} = \|\tilde{s} - s\| \quad \text{absolute error}$$

$$e_{rel} = \frac{\|\tilde{s} - s\|}{\|s\|} \quad \text{relative error}$$

- these are **forward errors** resulting from uncertainty in the data d due to FP representation
- other sources of uncertainty in data also possible

Absolute Condition Number

Assume $s = f(d) : \mathbb{R} \rightarrow \mathbb{R}$ has a Taylor series approximation around points of interest.

$$f(d + \Delta d) = f(d) + \Delta d f'(d) + O(\Delta d^2)$$

$$f(d + \Delta d) - f(d) \approx \Delta d f'(d)$$

$$|f(d + \Delta d) - f(d)| \leq |f'(d)||\Delta d|$$

$$e_{abs} = |\tilde{s} - s| \leq \kappa_{abs}|\Delta d|$$

Relative Condition Number

Assume $s = f(d) : \mathbb{R} \rightarrow \mathbb{R}$ has a Taylor series approximation around points of interest.

$$d + \Delta d = d(1 + \delta)$$

$$f(d + \Delta d) - f(d) \approx \Delta d f'(d)$$

$$\frac{|f(d + \Delta d) - f(d)|}{|f(d)|} \leq \frac{|f'(d)|}{|f(d)|}|\Delta d|$$

$$\frac{|f(d + \Delta d) - f(d)|}{|f(d)|} \leq \frac{|df'(d)|}{|f(d)|}|\frac{\Delta d}{d}|$$

$$\frac{|f(d + \Delta d) - f(d)|}{|f(d)|} \leq \frac{|df'(d)|}{|f(d)|}|\delta|$$

$$e_{rel} = \frac{|\tilde{s} - s|}{|s|} \leq \kappa_{rel}|\delta|$$

Condition Numbers

- Definitions generalize to multidimensional data and solutions.
- κ_{abs} bounds the e_{abs} with an amplification of the magnitude of the absolute change in the data.
- κ_{rel} bounds the e_{rel} with an amplification of the magnitude of the relative change in the data
- Gives an idea of how much the uncertainty in the data specifying the problem can change the solution, i.e., how many digits may be lost.
- This does not involve the precision of the finite arithmetic only the precision of representation.
- No algorithm is involved. This is an analytical statement.
- Problems with large condition numbers are called ill-conditioned.

A Simple Example

Consider the the condtioning of the sum of two numbers.

$$x = 151.72899 \text{ and } y = -151.71422$$

$$x + y = 0.01477$$

$$\tilde{x} = 151.73 \text{ and } \tilde{y} = -151.71$$

$$\tilde{x} + \tilde{y} = 0.02$$

- x and $\tilde{x}$ agree to 4 digits
- y and $\tilde{y}$ agree to 4 digits
- relative error in the two sums $0.35 = (0.02 - 0.01477)/0.01477$
- the respective sums do not agree to any digits

Sum is ill-conditioned when magnitudes are close but signs are opposite.

This is the cause of the numerical problem of cancellation.

Condition of Summation

$$s = \sum_{i=1}^{n} x_i \text{ and } \tilde{s} = \sum_{i=1}^{n} x_i(1+\eta_i), \qquad |\eta_i| \le \epsilon$$

$$\tilde{s} - s = \sum_{i=1}^{n} x_i \eta_i \rightarrow |\tilde{s} - s| \le \sum_{i=1}^{n} |x_i| \epsilon$$

$$|s| = |\sum_{i=1}^{n} x_i| \rightarrow \frac{|\tilde{s} - s|}{|s|} \le \kappa_{rel}\epsilon$$

$$\kappa_{rel} = \frac{|x_1| + \cdots + |x_n|}{|x_1 + \cdots + x_n|}$$

- $\kappa_{rel} = 1$ if all x_i have same sign.
- κ_{rel} large if the sum is small compared to the $|x_i|$, i.e., information is cancelled.

Conditioning

$$f(x) = e^{-x} \text{ and } f'(x) = -e^{-x}$$

$$\kappa_{rel} = \frac{|xe^{-x}|}{|e^{-x}|} = |x|$$

- Well-conditioned for $x = O(1)$.
- If $x \approx 10^k$ we lose k digits in $f(x)$ for every digit we alter in x.

Conditioning

$$x_0 = -5.5 \rightarrow e^{x_0} = 0.004086771\ldots$$

$$x_1 = -(5.5 + 10^{-5}) = x_0(1 + \delta_1), \quad \delta_1 = 1.8 \times 10^{-5}$$

expect agreement to 4 or 5 digits since $\quad |x_0||\delta_1| = 5.5 \times 1.8 \times 10^{-5}$

$$e^{x_1} = 0.00\underline{40867}31\ldots$$

Conditioning

$$x_2 = -(5.5 + 10^{-4}) = x_0(1 + \delta_2), \quad \delta_2 = 1.8 \times 10^{-4}$$

expect agreement to 3 to 4 digits since $\quad |x_0||\delta_2| = 5.5 \times 1.8 \times 10^{-4}$

$$e^{x_2} = 0.00\underline{4086}363\ldots$$

Conditioning

$$f(x) = \log x$$

$$f'(x) = \frac{1}{x}$$

$$\kappa_{rel} = \frac{|x|}{|x \log x|} = |\frac{1}{\log x}|$$

Ill-conditioned for $x \approx 1$.

Conditioning w/r to a Parameter

Given $p(x) = x^2 - 4x + \gamma$ consider roots.

$$x^2 - 4x + \gamma = 0 \rightarrow x_\pm = 2 \pm \sqrt{4-\gamma}$$

$$\gamma_0 = 4 \rightarrow x_\pm = 2$$

$$\gamma = \gamma_0 - 10^{-7} \rightarrow \tilde{x}_\pm = 2 \pm 10^{-3.5}$$

$$\Delta\gamma_{rel} = \frac{|\gamma - \gamma_0|}{|\gamma_0|} = \frac{10^{-7}}{4} = 2.5 \times 10^{-8}$$

$$\Delta x_{rel} = \frac{|\tilde{x} - x|}{|x|} = \frac{10^{-3.5}}{2} \approx 1.6 \times 10^{-4}$$

$$\frac{\Delta x_{rel}}{\Delta\gamma_{rel}} = \frac{1.6 \times 10^{-4}}{2.5 \times 10^{-8}} = 6400$$

Parameter agreed to 7 digits root agreed to 3 digits
loss of 4 digits characterized by lower bound on condition number of 6400

Conditioning w/r to a Parameter

Choice of parameterization matters. Given

$$p(x) = x^2 - 4x + \gamma_0 = x^2 - 4x + 4 = (x-2)^2$$

Suppose we view this as an instance of

$$\alpha(x - \rho_1)(x - \rho_2)$$

and discuss change in roots subject to parameters α, ρ_1, and ρ_2.

In this contrived parameterization the roots are perfectly conditioned.

See the text p. 35 and p. 39 for a more sophisticated version of this reparameterization with repeated roots.

Conditioning of Linear System Solving

Consider $Ax = b$

$$\begin{pmatrix} 0.300 & 0.401 \\ 0.374 & 0.500 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -0.101 \\ -0.126 \end{pmatrix}$$

$$\begin{pmatrix} 0.300 & 0.401 \\ 0.374 & 0.500 \end{pmatrix} \begin{pmatrix} -0.337 \\ 0 \end{pmatrix} = \begin{pmatrix} -0.1011 \\ -0.126038 \end{pmatrix}$$

$$\Delta A = 0, \Delta b = \begin{pmatrix} -1 \times 10^{-4} \\ -3.8 \times 10^{-5} \end{pmatrix}$$

original solution and solution to perturbed equation disagree in all digits!

Conditioning of Linear System Solving

Consider $Ax = b$

$$\begin{pmatrix} 0.3001 & 0.4012 \\ 0.374002 & 0.50004 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -0.1011 \\ -0.126038 \end{pmatrix}$$

$$\Delta A = \begin{pmatrix} 1 \times 10^{-4} & 2 \times 10^{-4} \\ 2 \times 10^{-6} & 4 \times 10^{-5} \end{pmatrix} \Delta b = \begin{pmatrix} -1 \times 10^{-4} \\ -3.8 \times 10^{-5} \end{pmatrix}$$

original solution and solution to perturbed equation are the same!

Conditioning of Linear System Solving

- The problem is ill-conditioned.
- We will define a condition number later.
- $\|A\|_\infty = 0.874$, $\|A^{-1}\|_\infty = 3.8 \times 10^4$
- Ill-conditioning does not imply that all perturbations to the data result in large changes to the solution.
- The condition number is a worst case bound on the effect of perturbations in a neighborhood around the original problem data.

Backward Error

- Conditioning relates perturbations in input data, d, to **FORWARD ERROR**

$$e_{rel} = \frac{\|f(d+\Delta d) - f(d)\|}{\|f(d)\|}$$

- conditioning addresses uncertainty analytically
- **BACKWARD ERROR** reverses the question.
- Given $s = f(d)$ and $\tilde{s} \neq f(d)$, find Δd such that

$$\tilde{s} = f(d+\Delta d)$$

- **That is, show that $\tilde{s}$ is the solution to another problem.**

Backward Error

- technique often shows existence of such a problem within some neighborhood of d.
- nearby problem may not exist in same “class” of problems
- turns stability analysis into a perturbation analysis