Errors associated with arithmetic operations

- To avoid subtraction between nearly equal numbers, consider

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}\left(\frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}}\right),$$

  which implies to

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}.$$

- We now get

$$x_1 = \frac{-2.000}{62.10 + 62.06} = -0.01610.$$

- The relative error $= 6.2 \times 10^{-4}$

Kinds of error and computer arithmetic

64-bit representation of a real number

The first bit is reserved for sign.

The following 11-bit is used for exponent, which gives a range of 0 to $2^{11} - 1 = 2047$. To ensure the representation of small number, ranges of exponent are $L = -1023$ and $U = 1024$.

Last 52-bit is used for mantissa, which corresponds to between 15 and 16 decimal digits.

In a normalized number system, the mentissa for the largest number is

$$\underbrace{1}.\underbrace{111\ldots\ldots1}_{52 \text{ digits}}.$$

Kinds of error and computer arithmetic

64-bit representation of a real number

$$x = \pm \left(d_0 + \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \dots \frac{d_{p-1}}{\beta^{p-1}} \right) \beta^E$$

$\pm$ is represented by $(-1)^s$, $p = 52$, and $\beta = 2$.

The first bit contains $s = 0$ to get a positive number.

For the mentissa of the smallest positive number, we get $d_0 = 1$ and $d_i = 0$ for all $1 \leq i \leq p - 1$.

In other words, mentissa is $1.0000\ldots\ldots.$

The smallest representable positive number is $(-1)^0(1+0)2^{-1022} \approx 0.2225 \times 10^{-307}$ - known as underflow.

Can we calculate the overflow (e.g. $(2 - 2^{-52})2^{1023}$)?

Numerical algorithm

An algorithm is a procedure that describes, in an unambiguous manner, a finite sequence of steps to be performed in a specified order.

The object of the algorithm is to implement a procedure to solve a problem or approximate a solution to the problem.

A description of the algorithm is called a pseudo-code that specifies the input to be supplied and the form of the output.

Numerical algorithm

Example:

Develop an algorithm for computing the Euclidean norm of an n-dimensional vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, which is defined by

$$\|\mathbf{x}\|_2 = \left[\sum_{i=1}^{n} x_i^2\right]^{1/2}.$$

Numerical algorithm

The pseudo-code of the algorithm:

Algorithm 1 Euclidean norm

```
INPUT n, x_1, x_2, ..., x_n
OUTPUT norm
  sum=0
  for  i = 1, 2, ..., n do
    sum = sum + x_i^2
  end for
  norm = sqrt(sum)
  return norm
```

Convergence and efficiency

Sensitivity and conditioning

- A problem is said to be insensitive, or well-conditioned, if a given relative change in the input data causes a reasonably commensurate relative change in the solution.
- A problem is said to be sensitive, or ill-conditioned, if the relative change in the solution can be much larger than that in the input data.
- A condition number is defined by

$$\mathrm{cond}(f) = \frac{|\,(f(\tilde{x}) - f(x))/f(x)\,|}{|\,(\tilde{x} - x)/x\,|} = \frac{|\,\Delta f/f\,|}{|\,\Delta x/x\,|} \approx \left|\, \frac{xf'(x)}{f(x)} \,\right|$$

Convergence and efficiency

Sensitivity and conditioning

Let $f(x) = \sqrt{x+1} - \sqrt{x}$ and x is large.

The concept of condition number implies that the calculation of $f(x)$ is stable.

How does one use the concept of condition number to explain subtraction cancellation error?

Submit assignment

All matlab code must be submitted using the submit assignment tool available in the Labnet system.

How does one use submit?

login to your linux labnet account.

Open a terminal (Application$\rightarrow$ Accessories$\rightarrow$terminal.

Submit assignment

mkdir amat3132

cd amat3132

mkdir amat3132-a1

copy files to amat3132-a1

Submit assignment

cd ~/amat3132

submit list

submit submit amat3132-1 a1

Use a1, a2, a3, a4, a5, a6 to specify the appropriate assignment.

Convergence and efficiency

Suppose that a numerical algorithm produces a sequence of approximations $x_1, x_2, x_3, \ldots$ that are approaching to the correct answer x^*. We say that the algorithm is convergent and write

$$\lim_{n \to \infty} x_n = x^*$$

if there corresponds to each positive ϵ a real number r such that $\mid x_n - x \mid < \epsilon$ whenever $n > r$. (n is an integer!).

Convergence and efficiency

Example: Since

$$\mid \frac{n+1}{n} - 1 \mid < \epsilon$$

whenever $n > \epsilon^{-1}$, then

$$\lim_{n \to \infty} \frac{n+1}{n} = 1.$$

Linear convergence: We say that the rate of convergence is at least linear if there is a constant $c < 1$ and an integer N such that

$$\mid x_{n+1} - x^* \mid \leq c \mid x_n - x^* \mid \quad (n \geq N).$$

Convergence and efficiency

Super-linear convergence: We say that the rate of convergence is at least super-linear if there exist a sequence ϵ_n tending to 0 and an integer N such that

$$\mid x_{n+1} - x^* \mid \leq \epsilon_n \mid x_n - x^* \mid \quad (n \geq N).$$

Quadratic convergence: We say that the rate of convergence is at least quadratic if there exist C and an integer N such that

$$\mid x_{n+1} - x^* \mid \leq C \mid x_n - x^* \mid^2 \quad (n \geq N).$$

Convergence and efficiency

Big $\mathcal{O}$: Let x and y be two different numbers that depend on the parameter ϵ. If there are constants C and ϵ^* such that $\mid x \mid \leq C \mid y \mid$ if $\epsilon \to \epsilon^*$, then we write

$$x = \mathcal{O}(y), \quad \epsilon \to \epsilon^*.$$

Linear convergence: $\mid x_{n+1} - x^* \mid \leq \mathcal{O}(\mid x_n - x^* \mid) \quad (n \geq N).$

Super-linear convergence: $\mid x_{n+1} - x^* \mid \leq \mathcal{O}(\mid x_n - x^* \mid) \quad (n \geq N).$

Quadratic convergence: $\mid x_{n+1} - x^* \mid \leq \mathcal{O}(\mid x_n - x^* \mid^2) \quad (n \geq N).$

Submit assignment

All matlab code must be submitted using the submit assignment tool available in the Labnet system.

How does one use submit?

login to your linux labnet account.

Open a terminal (Application$\rightarrow$ Accessories$\rightarrow$terminal.

Submit assignment

mkdir amat3132

cd amat3132

mkdir amat3132-a1

copy files to amat3132-a1

Submit assignment

cd ~/amat3132

submit list

submit submit amat3132-1 a1

Use a1, a2, a3, a4, a5, a6 to specify the appropriate assignment.

System of linear equations

A set of simultaneous linear algebraic equations can be expressed as

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_2
\end{aligned}$$

where n is the number of unknowns, the coefficients a_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, n$ and the constants b_i, $i = 1, \ldots, n$ are known, and x_i, $i = 1, \ldots, n$ are unknowns.

System of linear equations

This system can also be written as

$$A\mathbf{x} = \mathbf{b},$$

where A is the $n \times n$ coefficient matrix, and b, $\mathbf{x}$ are vectors of size n.

The solution methods can be divided into two types:

1. Direct methods.
2. Indirect or iterative methods.

System of linear equations

Commonly used direct methods:

1. Gauss elimination method.
2. Gauss-Jordan method.
3. LU decomposition method.

System of linear equations

Commonly used iterative methods:

1. Jacobi method.
2. Gauss-Seidel method.
3. Relaxation method.

Direct method

Example:

Let us consider a linear system $Ax = b$, the matrix A and the right hand side vector b is given such that the augmented matrix $A \mid b$ can be written as:

$$\left[\begin{array}{rrr:r} 4 & -2 & 1 & 15 \\ -3 & -1 & 4 & 8 \\ 1 & -1 & 3 & 13 \end{array}\right]$$

Direct method

Solution:

We want to solve for x using the Gaussian elimination technique.

After elimination, the augmented matrix takes the following form:

$$\left[\begin{array}{rrr:r} 4 & -2 & 1 & 15 \\ 0 & -2.5 & 4.75 & 19.25 \\ 0 & 0 & 1.80 & 5.40 \end{array}\right]$$

Direct method

Clearly, we can solve the last equation.

Therefore, start from the last equation, and move towards the first equation.

Using the eliminated augmented matrix, we get

$x_3 = 3$

$x_2 = -2$

$x_1 = 2$

The process is known as back substitution.