- alternative characterization, ε_{mach} smallest # s.t.

fl(1 + eps_mach) > 1

Examples:
- (Ex. 1) eps_mach (chop, nearest) = .25, .125
- IEEE SP eps_mach (nearest) = 2^-24 ~= 10^-7 (about 7 decimal digits of precision)
- IEEE DP eps_mach (nearest) = 2^-53 ~= 10^-16 (about 16 decimal digits of precision)

Floating Point Math

+, −

- adding or subtracting +, −
  - match exponents first
  - must shift smaller number
  - if the sum (or diff) contains more than p digits, then the ones smaller than p will be lost
  - smallest number may be lost completely

×

- multiplication ok
  - mult mantissas and sum exponents
  - still need to round though, because product will generally have more digits (up to 2p)
- division (also need to round)

Example

at this point smaller # totally lost

```
    1.23 * 10^5
+   1.00 * 10^4 (10^3, 10^2)
```

$\begin{pmatrix} 1.23 \times 10^5 \\ + 1.00 \times 10^4 \end{pmatrix} \Rightarrow \begin{pmatrix} 1.23 \times 10^5 \\ .10 \times 10^5 \end{pmatrix}$ 1.33×10^5

$\begin{pmatrix} 1.23 \times 10^5 \\ 1.00 \times 10^2 \end{pmatrix} \Rightarrow \begin{pmatrix} 1.23 \times 10^5 \\ 0.00 \times 10^5 \end{pmatrix}$ (1.23×10^5)

- can also get overflow or underflow
- underflow often ok - 0 is good approximation
- overflow more serious problem - can't approximate the number in question

- IEEE standard gives us $op = +, -, \times, \div$

```
x flop y = fl(x op y)
```

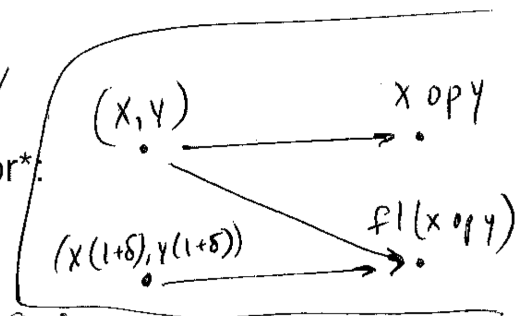
as long as overflow doesn't occur
- + and * commutative but *not* associative
- Ex: for eps < eps_mach, and 2 eps > eps_mach

```
  ( 1 + eps ) + eps  = 1
    1 + ( eps + eps )  = 1 + 2 eps > 1
```

Rounding Error Analysis

Basic idea is:

```
    fl(x op y) = (x op y)(1 + delta),
        |delta| <= eps_mach, and op = +, -, *, /
```

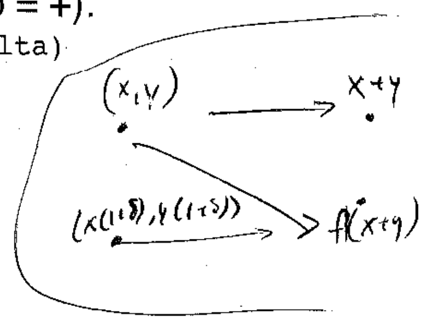


rearranging, get bound on relative *forward error*:

```
|fl(x op y) - (x op y)|
-----------------------  = |delta| <= eps_mach
     |(x op y)|
```

or, can interpret in terms of *backward error* (with op = +):

```
fl(x + y) = (x + y)(1 + delta) = x(1+delta) + y(1+delta)
```




```
Example: Compute x(y+z)
--------

fl(y+z) = (y+z)(1+d1), |d1|<=eps_mach
and
fl(x(y+z)) = (x(y+z)(1+d1))(1+d2), |d2|<=eps_mach
           = x(y+z)(1+d1+d2+d1d2)
         ≈ ~= x(y+z)(1+d1+d2)
           = x(y+z)(1+d), |d| = |d1 + d2| <= 2 eps_mach
```

- pessimistic bound
- typical, multiples of eps_mach accumulate
  - but in practice this is generally ok

Cancellation

problems can arise when subtracting two very close numbers
- result is exactly representable, but
- e.g., if the numbers differ by rounding error, this can basically leave rounding error only after subtracting

```
Examples
--------

  x = 1.92403 * 10^2
- y = 1.92275 * 10^2
  -----------------
      0.00128 * 10^2  = .128 = 1.28 * 10^-1
```

- only 3 significant digits in the result

BAD: computing *small quantity* as a difference of *large quantities*

```
e^x = 1 + x + x^2/2 + x^3/3! + ..., for x < 0
```

```
Example:  Quadratic formula
-------

ax^2 + bx + c = 0

     -b +- sqrt(b^2-4ac)
b =  -------------------
             2a

0.05010 x^2 - 98.78 x + 5.015
roots ~= 1971.605916,   answer to 10 digits
          0.05077069387
```

```
b^2 - 4ac = 9757-1.005 = 9756  answer to 4 digits
sqrt( " ) = 98.77
roots: (98.78 +- 98.77) / 0.1002 = 1972, 0.09980
```

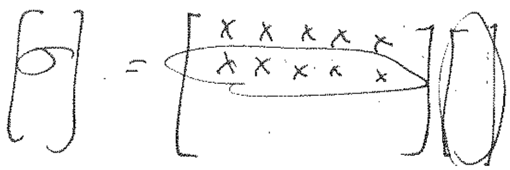
subtraction of two *close* numbers (cancellation error), followed by division by *small* number (amplification)

① Intro (T&B, Lecture 1) $\boxed{\text{Lecture 2}}$

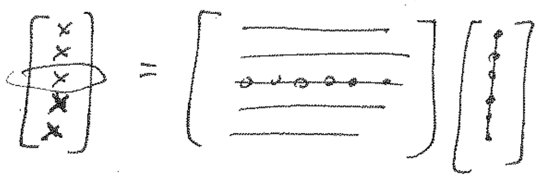
matrix	A	vector x	vector b	Ax mat-vec
entries	a_{ij}	x_i	b_i	αx s-v

② A linear map $\boxed{b = Ax}$

③ $\boxed{\text{Matrix-Vector Multiplication}}$ $\cancel{b_i = \sum_{j=1}^{n} a_{ij} x_j}$



"row" view of matrix multiplication



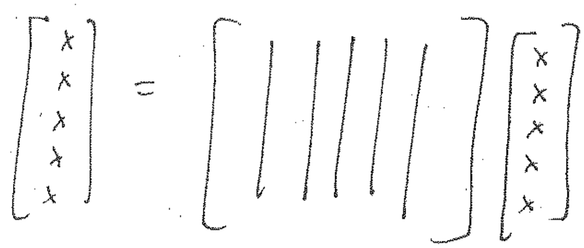
"row view" of matrix multiplication

$$b \quad = \quad A \quad x$$

$$\boxed{b_i = \sum_{k=1}^{n} a_{ik} x_k}$$

"row view"

④



"column view"

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\vec{b} = \sum_{j=1}^{n} x_j \vec{a}_j = x_1 \vec{a}_1 + x_2 \vec{a}_2 + \cdots + x_n \vec{a}_n$$

$\vec{b}$ is linear combination of $\vec{a}_j$

Ex. Vandermonde Matrix

Matrix - Matrix

$$\cancel{B = CA} \qquad B = AC$$

$$b_{ij} = \sum_{k=1}^{n} a_{ik} c_{kj}$$

$$\vec{b}_j = \sum_{k=1}^{n} \vec{a}_k \, c_{kj}$$

Ex. Outer product

$$\vec{u}\,\vec{v}^T$$

$$\begin{bmatrix} | \\ \vec{u} \\ | \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} | & | & & | \\ v_1\vec{u} & v_2\vec{u} & \cdots & v_n\vec{u} \\ | & | & & | \end{bmatrix}$$

$$= \begin{bmatrix} v_1u_1 & v_2u_1 & & v_nu_1 \\ v_1u_2 & v_2u_2 & & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ v_1u_n & v_2u_n & & v_nu_n \end{bmatrix}$$

Ex upper triangular, U

$$B = AU = \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} \begin{bmatrix} \ddots & & \\ & \ddots & \\ & & \ddots \end{bmatrix}$$

$$\cancel{b_k} \qquad b_j = \sum_{k=1}^{j} a_{k}$$

Range + Nullspace

range : set of vectors that can be expressed as

"column space" Ax

i.e. space ~~of~~ spanned by columns of A.

nullspace . vectors x ~~the~~ s.t.

$$Ax = \vec{0}$$

rank : dim (col space)

inverse

$$A^{-1}A = AA^{-1} = I.$$