**linear system**: central problem of linear algebra

* **linear equation**: *a1x1 + a2x2 + … + axnx = b*
  + **a** is coefficient;
    - **a** and **b** can be complex(real) numbers
  + **x** is unknown
* **linear system**: a collection of linear equation**s**
  + solution: makes each equation a true statement
  + vector equation is the equivalent of a linear system
    - **singular case**
      * no solution
      * infinitely many solutions
    - **nonsingular case:** exactly one solution
      * number of **equations** = number of **unknowns**
* **two solution ways**
  + **elimination:** gaussian elimination
  + **determinants:** not suitable to solve large system
* **geometry aspect:** parallel and intersect
  + **nonsingular case**
    - **row picture(Intersection of planes)**
      * each equation:**(n-1) dimensional plane** in **n dimensions**
      * one equation reduces the dimension by one.
      * **goal**: intersect at a point[**0** dimension]
    - **column picture(combination of columns)**
      * ask for a linear combination of **n** columns that equals b
      * **relation** to row picture
        + **vector** of multipliers of combination is the **point** of intersection in row picture
  + **singular case**
    - **row picture:** 
      * **fail**: dimension of resulting intersection > 0
      * **infinity of solutions**
        + parallel or parallel after reducing
      * **inconsistent**
        + no solution: non-intersect
    - **column picture:**
      * **fail**: vectors can’t extend to a n dimensions
        + some column vectors in the same plane

three columns lie in the same plane

combination equals 0

* + - * **infinity of solutions**
        + b **in** that lower dimensional space
      * **inconsistent**
        + no solution: b **not in** that lower dimensional space
* **gaussian elimination(**a full set of n pivots = one solution**)**
  + calculate along right side (b)
  + **subtracting** **multiples of the equation** from the **equations beneath**
    - **specific steps**
      * **forward elimination**
        + **1)** constantly **dividing** the **pivot** into the numbers underneath it, to find out the **right** **multipliers**.
        + **2)** **determine** every **pivot**(**=! 0**), by **eliminating** cooresponding **pivot** from **equations beneath**
      * **back-substitution** in reverse order
        + the last **pivot** can solve the last linear equation
  + **Breakdown(a zero appears in a pivot position)**
    - **singular case**(permanently stop)
      * **a full set of pivots** cannot be found
    - **non singular case**(temporarily stop)
      * **cured by** exchanging equations
        + have no solution
        + infinitely many
  + **Cost**
    - **forward elimination**
      * **division:** *(n-1)+(n-2)+...+1 = (n-1+1)(n-1)/2 =* ***n(n-1)/2***
      * **multiplication with subtraction:**
        + *n\*(n-1)+(n-1)\*(n-2)+...+1 = n(n+1)(2n+1)/6 - n(n+1)/2 =* ***(n^3-n)/6***

n is large: ***⅓\*n^3***

* + - **back-substitution and right division**
      * **back-substitution: *n^2***
        + **division/substitution/subtraction:** *1+2+...+n =* ***n(n+1)/2***
        + **right division:** *(n-1)+(n-2)+...+1 = (n-1+1)(n-1)/2 =* ***n(n-1)/2***
    - The newest problem is the cost with many processors in parallel
* **triangular Factors**: gaussian elimination in terms of Matrics
  + **vector equation**: ***A****x=b* [***A***: matrix *x*: vector *b*: vector]
    - **coefficient matrix: *A***
    - **augmented matrix: *A*** augmented by *b*
  + **elementary matrix** (adjusted identity matrix)
    - according to the meaning of matrics multiplication, change corresponding entry
  + **Triangular factorization *A* = *LU*** ***U****x* = *c*
    - No exchanges of rows
    - **U**: **upper triangular matrix**
      * **appears:** after forward elimination
      * **diagonal: the pivots**
    - ***L***: **lower triangular matrix**
      * ***L*** applied to ***U*,** brings back ***A***
      * **diagonal: 1s**
      * **below the diagonal:** multipliers
    - ***L*** and ***U*** are **unsymmetric**
  + **other way**
    - ***A* = *LDV*(*LDU*)**
      * ***L*** and ***V***: 1s on the diagonal(**unit diagonal**)
        + ***A* = *LDLT V*** is ***LT*** : symmetric products

if ***A*** is symmetric

step can be reduced from n3/3 to n3/6

* + - * ***D***: the diagonal matrix of pivots()
        + **no zeros** on the diagonal
    - ***LDU*** and ***LU*** factorization are uniquely determined by ***A***.
  + **One linear system = two triangular systems**
    - **factor**(from ***A*** find its factors ***L*** and ***U***)
    - **solve**(from ***L*** and **U** and *b* find **the solution** *x*)
      * ***L****c = b*: is solved forward
      * ***U****x = c*: is solved backward
    - **two triangular systems** in *(n^2)/5* steps
    - a series of b’s can be processed
      * can be found in *n^2* operations
  + **zero in the pivot location**
    - **curable:** **nonzero** entry lower down in the same column **exist**
      * **a row exchange** is carried out
      * **the nonzero entry** becomes the **needed pivot**
    - **incurable:** **nonzero** entry lower down in the same column **inexist**
      * **singular**
  + **elimination in a nutshell: *PA = LU***
    - **firstly: rows reordering**(**avoid** 0 in **pivot position**)
      * **reduce roundoff error**: exchange the pivot that is near zero
    - **secondly: *PA***can be factored into ***LU***