Another class of problems where round-off errors come into play are ill-conditioned problems.

Here the algorithm is fine, the computation of the answer is precise, but the answer is wrong!

$\Rightarrow$ Because of round-off error and the nature of the problem, you are solving a slightly different problem exactly!

Suppose we want the roots of a polynomial:

$$\left(x - \tfrac{2}{3}\right)^4 = 0$$

This has the repeated root

$$x = \tfrac{2}{3}$$

Now suppose we solve this by attacking the polynomial when multiplied out:

$$x^4 - 4\tfrac{2}{3}x^3 + 6\tfrac{4}{9}x^2 - 4\tfrac{8}{27}x + \tfrac{16}{81} = 0$$

and we store all #'s to single precision. (floating pt. rep.)

If we solve the resulting problem, we get:

0.6861
0.6663 + 0.0191 i ($i \equiv \sqrt{-1}$)
0.6663 − 0.0191 i
0.6480

or an error of 2.8%!
(even w/ double precision error is ~0.01%)

Why? problem was <u>ill</u> <u>conditioned</u>

The computer solved a problem which was <u>very</u> similar to the one we wanted, but had a very different answer!

Let's see what happens if we plug one of the roots back in:

$$x = 0.6861, \quad \left(x - \frac{2}{3}\right)^4 = 1.4 \times 10^{-7}$$

which is within roundoff error of zero! (error for double was 1.8×10^{-17})

In an ill-conditioned problem, a small change in the coefficients yields a large change in the result.

The difficulty is in the problem, not the algorithm

Try to avoid ill-conditioned problems as much as possible.

A constant focus of the course will be how to avoid all of these types of numerical errors.

Systems of Linear Equations

We all know that 2pts determine a line, 3pts a parabola, 4pts a cubic, etc. How do we get the curves?

Say we have 3pts:

$(1,0)\ ,\ (2,-1)\ ,\ (3,2)$

We want to determine the coefficients of the parabola:

$$p(x) = a + bx + cx^2$$

Just plug in the points!

$$p(1)=0,\quad p(2)=-1,\quad p(3)=2$$

So:
$$\begin{aligned} a + b + c &= 0 \\ a + 2b + 4c &= -1 \\ a + 3b + 9c &= 2 \end{aligned}$$

We can write this in matrix form:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 2 \end{pmatrix}$$

How do we solve this? This is of the form:

$$\underset{\approx}{A}\, \underset{\sim}{x} = \underset{\sim}{b}$$

where the coef.'s $\underset{\sim}{x}$ are unknown!

This is a <u>system</u> of linear equations!

We <u>can</u> solve by inverting $\underset{\approx}{A}$:

let $\underset{\approx}{A}^{-1}$ be a matrix s.t.

$$\underset{\approx}{A}^{-1} \underset{\approx}{A} = \underset{\approx}{I} \quad \text{(identity matrix)}$$

Then:

$$\underset{\approx}{A}^{-1} \underset{\approx}{A}\, \underset{\sim}{x} = \underset{\approx}{A}^{-1} \underset{\sim}{b}$$

$$\text{or } \underset{\sim}{x} = \underset{\approx}{A}^{-1} \underset{\sim}{b}$$

This is <u>correct</u>, but <u>inefficient</u> since we don't need to invert $\underset{\approx}{A}$ to solve the problem! Use Gaussian elimination instead.

How does this work? The same way you would naturally do it!

Say:
$$x_1 + x_2 + x_3 = 0$$
$$x_1 + 2x_2 + 4x_3 = -1$$
$$x_1 + 3x_2 + 9x_3 = 2$$

Subtract 1st eq'n from second 2, elim. x_1 from eq'ns. Thus:

$$\begin{aligned} x_1 + x_2 + x_3 &= 0 \\ x_2 + 3x_3 &= -1 \\ 2x_2 + 8x_3 &= 2 \end{aligned}$$

Now subtract twice 2nd eq'n from third:

$$\begin{aligned} x_1 + x_2 + x_3 &= 0 \\ x_2 + 3x_3 &= -1 \\ 2x_3 &= 4 \end{aligned}$$

This is the new problem:

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 4 \end{pmatrix}$$

Which is in the upper triangular form:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & & \vdots \\ 0 & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ \vdots \\ c_n \end{pmatrix}$$

This can be solved by back substitution

$$x_i = \begin{cases} c_n / a_{nn} \\ \left(c_i - \sum_{j=i+1}^{n} a_{ij} x_j\right) / a_{ii} & i < n \end{cases}$$

For this case,

$$x_3 = 2, \quad x_2 = -7, \quad x_1 = 5$$

How much time does this take?

We require $\sim n^3/3$ multiplications for Gaussian elimination

Back substitution takes $\sim n^2$ operations

$\therefore$ dominated by elimination!

What does this mean?

Suppose we have 100×100 matrix

$\rightarrow$ need $O(100^3) \sim 10^6$ operations

How about a 1000×1000 system?

need 1000 times longer!

A <u>big</u> problem in large scale simulations

Often matrices are <u>sparse</u> or have a banded structure that allows the use of special algorithms $\Rightarrow$ faster

A matrix is *sparse* if only a few elements are non-zero

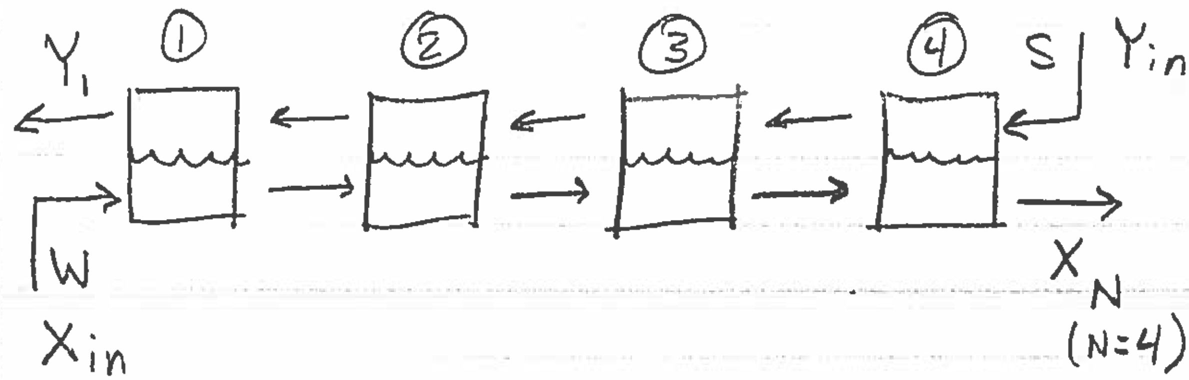
A matrix is *banded* if non-zero elements are along the diagonals

Special algorithms, faster than $O(n^3)$ can be used to solve these types of problems!

Let's examine a Chem Eng. example:

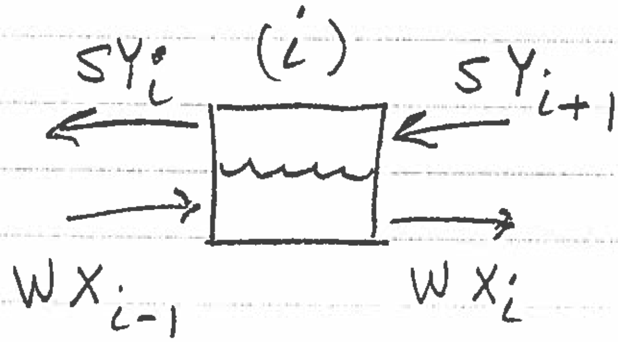
Liquid-Liquid Extraction

We have a four-stage extractor:



We are stripping a dilute solute from a water stream W by contacting this stream with a solvent stream S. The solute concentration in the water stream is X, and in the solvent stream is Y. We want to determine the outlet concentration of the water stream X_N ($N=4$ in this case).

In each stage we have a mass balance:



The solute in = solute out, thus:

$$W X_{i-1} + S Y_{i+1} = W X_i + S Y_i$$

We also have the equilibrium relation (Henry's Law):

$$Y_i = K X_i$$

Combining these two we get the relation for the interior stages:

$$W X_{i-1} + S K X_{i+1} = (W + SK) X_i$$

or, dividing by W and rearranging:

$$X_{i-1} - \left(1 + \frac{SK}{W}\right) X_i + \frac{SK}{W} X_{i+1} = 0$$

We need different equations for the first and last contactors:

$$W X_{in} + SK X_2 = (W + SK) X_1$$

or

$$-\left(1 + \frac{SK}{W}\right) X_1 + \frac{SK}{W} X_2 = -X_{in}$$

for the first contactor, and:

$$W X_{N-1} + S Y_{in} = (W + SK) X_N$$

or

$$X_{N-1} - \left(1 + \frac{SK}{W}\right) X_N = -\frac{S}{W} Y_{in}$$

So the problem depends on X_{in}, Y_{in} and the key ratio:

$$\chi \equiv \frac{SK}{W}$$

Let's take another look at this example

we have the equation for the first contactor:

$$-(1+\chi)x_1 + \chi x_2 = -x_{in}$$

In matrix form we get:

$$\begin{pmatrix} -(1+\chi) & \chi & 0 & \cdots & 0 \\ & & & & \\ & & & & \\ & & & & \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} -x_{in} \\ \vdots \\ \\ \end{pmatrix}$$

Now we add in the add'l eq'ns:

$$x_{i-1} - (1+\chi)x_i + \chi x_{i+1} = 0$$

and the last eq'n

$$x_{N-1} - (1+\chi)x_N = -\chi \frac{Y_{in}}{K}$$

thus:

$$\begin{pmatrix} -(1+\chi) & \chi & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -(1+\chi) & \chi & 0 & 0 & \cdots & & \vdots \\ 0 & 1 & -(1+\chi) & \chi & 0 & \cdots & & 0 \\ & \ddots & & \ddots & & \ddots & & \chi \\ & 0 & & & 0 & & 1 & -(1+\chi) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} -x_{in} \\ 0 \\ \vdots \\ 0 \\ -\chi \frac{Y_{in}}{K} \end{pmatrix}$$